UNIVERSITY OF CALIFORNIA, SAN DIEGO

Duplications and Genome Rearrangements

A dissertation submitted in partial satisfaction of the
requirements for the degree Doctor of Philosophy

in

Computer Science

by

Max Alekseyev

Committee in charge:

Professor Pavel Pevzner, Chair
Professor Vineet Bafna
Professor Fan Chung Graham
Professor Russell Impagliazzo
Professor Glenn Tesler

2007

.

The dissertation of Max Alekseyev is approved, and it is acceptable in quality and form for publication on microfilm:

_____

_____

_____

_____

_____

Chair

University of California, San Diego

2007

To Angelica and Daniel.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

First and foremost, I am immensely grateful to my advisor, Professor Pavel Pevzner for his guidance and financial support throughout my Ph.D. journey. I feel fortunate and privileged to have studied with him, from whom I learned not only the fascinating area of computational biology and the art of research but also such basic things as English writing style and clear exposition of the material in research papers. His insightful intuition helped me to choose a right direction of research, to come up with non-trivial new results, and to simplify some "ugly" looking results into an elegant form. I proudly share with Pavel the discovery of many results present in this dissertation. I was always (and still is) impressed by Pavel's encyclopedic knowledge and ability to easily explain difficult topics to literally anybody. Despite these many years we worked together, there is still a lot of things that I would like to learn from Pavel.

I feel obliged to Professor Glenn Tesler for a number of extensive reviews of my papers that he has kindly provided. I also thank Glenn for numerous insightful discussions and hope to write a joint paper in future.

I wish to thank Professors Vineet Bafna, Mihir Bellare, Fan Chung Graham, Ronald Graham, T. C. Hu, Russell Impagliazzo, Daniele Micciancio, Pavel Pevzner, and Van Vu whose remarkable lectures I found the most helpful and enjoyable among the courses that I was taking during my Ph.D. study at UCSD.

A special "thank you" goes to Professors Vineet Bafna, Russell Impagliazzo, Fan Chung Graham, Pavel Pevzner, and Glenn Tesler for taking their time to review my dissertation and serve on my defense committee.

I thank my fellows, former and current graduate students and postdocs in the UCSD CSE Bioinformatics Lab, for interdisciplinary, friendly and fun learning environment, which has made my Ph.D. study a very precious and unique experience. Among many others, I would like to thank Steffen Heber, Zufar Mulyukov, Ben Raphael, Haixu Tang, Shaojie Zhang, Degui Zhi, Neil Jones, Nuno Bandeira, Mark Chaisson, and Qian Peng.

I thank my former research advisors Professors Yevgeniy I. Gordon and Valeriy N. Shevchenko at Nizhni Novgorod State University, Russia. My scientific achievements will always be a part of yours.

Finally, I am deeply indebted to my wife Angelica and son Daniel for their everlasting support and love.

Chapter 2 is based on the following three papers:

- Max A. Alekseyev and Pavel A. Pevzner. "Multi-Break Rearrangements and Chromosomal Evolution". *Theoretical Computer Science*, 2007. (to appear)
- Max A. Alekseyev and Pavel A. Pevzner. "Are There Rearrangement Hotspots in the Human Genome?". *PLoS Computational Biology*, 2007. (to appear)
- Max A. Alekseyev. "Multi-Break Rearrangements: from Linear to Circular Genomes". *Proceedings of the 5th Annual RECOMB Satellite Workshop on Comparative Genomics*, 2007. (to appear)

Chapter 3 is based on the following three papers:

- Max A. Alekseyev and Pavel A. Pevzner. "Whole Genome Duplications and Contracted Breakpoint Graphs". *SIAM Journal on Computing*, 2007, 36(6), pp. 1748-1763.
- Max A. Alekseyev and Pavel A. Pevzner. "Colored de Bruijn Graphs and the Genome Halving Problem". *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2007, 4, pp. 98-107.
- Max A. Alekseyev and Pavel A. Pevzner. "Whole Genome Duplications, Multi-Break Rearrangements, and Genome Halving Theorem". *Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2007, pp. 665-679.

The dissertation author was the primary investigator and author of these papers.

VITA

1997	B.S. in Mathematics (*summa cum laude*) Nizhni Novgorod State University, Nizhni Novgorod, Russia
1999	M.S. in Mathematics (*summa cum laude*) Nizhni Novgorod State University, Nizhni Novgorod, Russia
2007	Ph.D. in Computer Science University of California, San Diego

PUBLICATIONS

[Ale07] Max A. Alekseyev. Multi-Break Rearrangements: from Circular to Linear Genomes. *Proceedings of Fifth Annual RECOMB Satellite Workshop on Comparative Genomics, September 14-16, La Jolla, California (Lecture Notes in Bioinformatics)*, 2007. (to appear).

[AP04] Max A. Alekseyev and Pavel A. Pevzner. Genome Halving Problem Revisited. *Lecture Notes in Computer Science*, 3328:1–15, 2004.

[AP07a] Max A. Alekseyev and Pavel A. Pevzner. Are There Rearrangement Hotspots in the Human Genome? *PLoS Computational Biology*, 2007. (to appear).

[AP07b] Max A. Alekseyev and Pavel A. Pevzner. Colored de Bruijn graphs and Genome Halving Problem. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 4(1):98–107, 2007.

[AP07c] Max A. Alekseyev and Pavel A. Pevzner. Multi-Break Rearrangements and Chromosomal Evolution. *Theoretical Computer Science*, 2007. (to appear).

[AP07d] Max A. Alekseyev and Pavel A. Pevzner. Whole Genome Duplications and Contracted Breakpoint Graphs. *SIAM Journal on Computing*, 36(6):1748–1763, 2007.

[AP07e] Max A. Alekseyev and Pavel A. Pevzner. Whole Genome Duplications, Multi-Break Rearrangements, and Genome Halving Theorem. *Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 665–679, 2007.

[HAS+02] Steffen Heber, Max Alekseyev, Sing-Hoi Sze, Haixu Tang, and Pavel Pevzner. Splicing graphs and EST assembly problem. *Bioinformatics*, 18(S1):181–188, 2002.

ABSTRACT OF THE DISSERTATION

Duplications and Genome Rearrangements

by

Max Alekseyev
Doctor of Philosophy in Computer Science
University of California, San Diego, 2007
Professor Pavel Pevzner, Chair

Most genome rearrangements (e.g., reversals and translocations) can be represented as *2-breaks* that break a genome at 2 points and glue the resulting fragments in a new order. Multi-break rearrangements break a genome into multiple fragments and further glue them together in a new order. While multi-break rearrangements were studied in depth for $k = 2$ breaks, the k-break distance problem for arbitrary k remains unsolved. In the first part of this dissertation we address several open issues and problems related to the multi-break rearrangements:

1. We prove a duality theorem for the multi-break distance problem between circular genomes and give a polynomial algorithm for computing this distance.

2. In 2003 Pevzner and Tesler [56] refuted the Random Breakage Model (RBM), that had been a de-facto theory of molecular evolution for more than three decades, and introduced a new Fragile Breakage Model (FBM). However, the rebuttal of RBM caused a controversy and led to a split among researchers studying genome evolution. In particular, since the mathematical theory used to refute RBM does not cover more complex rearrangement operations (like transpositions), the Pevzner–Tesler arguments do not apply for the case when transpositions are frequent. We contribute to the ongoing debates on "RBM vs. FBM" controversy by analyzing multi-break rearrangements and demonstrating

that even if transpositions were a dominant force in mammalian evolution, the arguments in favor of FBM still stand.

3. We extend the above results to the more difficult case of linear genomes. In particular, we give lower bounds for the rearrangement distance between linear genomes and use these results to analyze comparative genomic architecture of mammalian genomes.

In the second part of this dissertation we study rearrangements in genomes with duplicated genes. In particular, we focus on the Genome Halving Problem motivated by the whole genome duplication events in molecular evolution, first formulated and solved by Nadia El-Mabrouk and David Sankoff. We propose a new approach to analysis of rearrangements in genomes with duplicated genes, based on a generalization of conventional breakpoint graph, that let us to obtain the following results:

1. We reformulate the problem of computing the rearrangement distance between genomes with duplicated genes as two graph-theoretical problems and demonstrate how to solve their particular variants appearing in the course of solving the Genome Halving Problem. For the Genome Halving Problem with 2-breaks (i.e., standard rearrangements) this leads to an alternative short solution.

2. We further study the Genome Halving Problem with 3-breaks that add transpositions to the set of standard rearrangement operations considered by El-Mabrouk and Sankoff [24]. The El-Mabrouk–Sankoff analysis of the 2-Break Genome Halving Problem is already rather complex making it appears unlikely that there exists a similar result for the 3-Break Genome Halving. We prove the contrary by giving a polynomial algorithm and duality theorem for the 3-Break Genome Halving Problem.

3. We reveal that while the El-Mabrouk–Sankoff analysis of the Genome Halving Problem is correct in most cases, it does not hold in the case of unichromosomal circular genomes. This raises a problem of correcting the El-Mabrouk–Sankoff

analysis and devising an algorithm that deals adequately with all genomes. We efficiently classify all genomes into two classes and show that while the El-Mabrouk–Sankoff theorem holds for the first class, it is incorrect for the second class. The crux of our analysis is a new combinatorial invariant defined on duplicated permutations. Using this invariant we were able to come up with a full proof of the Genome Halving theorem and a polynomial algorithm for the Genome Halving Problem (for unichromosomal circular genomes).

1 Introduction

In 1970 Susumu Ohno came up with two fundamental theories of chromosome evolution that were subjects to many controversies in the last 35 years [49]. One of them, *Whole Genome Duplication Model*, was first met with scepticism and only recently was proven to be correct [37, 19]. The other, *Random Breakage Model*, had a very different fate. It was embraced by biologists from the very beginning and only recently was refuted by Pevzner and Tesler, 2003 [56]. In this dissetation we study computational problems related to the Whole Genome Duplication Model and the Random Breakage Model in presence of complex rearrangements (e.g., transposition), and develop a general technique for analyzing such rearrangements.

1.1 Random Breakage Model vs. Fragile Breakage Model of Chromosomal Evolution

Rearrangements are genomic "earthquakes" that change the chromosomal architectures. The fundamental question in molecular evolution is whether there exist "chromosomal faults" where rearrangements are happening over and over again, resulting in high *breakpoint reuse rate*. The *Random Breakage Model (RBM)*, proposed by Ohno [49] and formalized by Nadeau and Taylor [48], postulates that rearrangements happen at "random" genomic positions, thus implying low breakpoint reuse rate across mammalian genomes. Because of its prophetic prediction power, RBM became the *de facto* theory of chromosomal evolution. Only recently Pevzner and Tesler, 2003 [56] refuted RBM and suggested an alternative *Fragile Breakage Model*

(FBM) of chromosomal evolution. The FBM postulates existence of *fragile* genomic regions that are more likely to be broken by rearrangements than the rest of the genome, implying (in contrast to the RBM) high breakpoint re-use rate. A variety of further studies argued for existence of fragile regions in mammalian genomes [47, 70, 4, 78, 74, 33, 58, 77, 44, 38]. For example, Kikuta et al, 2007 [38] analyzed the links between genome fragility and the need to keep genome intact by regulatory elements and came to the conclusion that *"the Nadeau and Taylor hypothesis is not possible for the explanation of synteny in general."*

At the same time the rebuttal of RBM caused a controversy and was followed by extensive debates [63, 64, 52, 62] on the validity of Pevzner–Tesler's arguments. In particular, Sankoff, 2006 [62] questioned the assumption adopted by Pevzner and Tesler [56] that chromosomal architectures mainly evolve by the "standard" rearrangement operations (i.e., reversals, translocations, fissions, and fusions). Indeed, since the mathematical theory used to refute RBM does not cover more complex rearrangement operations (like transpositions), the arguments in [56] do not apply for the case when transpositions are frequent.

In this dissertation we develop a theory for analyzing complex rearrangements (including transpositions) and demonstrate that even if such rearrangements were a dominant evolutionary force, there are still rearrangement hotspots in mammalian evolution.

1.2 Whole Genome Duplications

The Whole Genome Duplication Model postulates a new type of evolutionary events that duplicate each chromosome of a genome. It had been a subject to controversy for the first 35 years [49, 75, 65, 9, 40, 39] and only recently was proven to be correct [37, 19]. Kellis et al., 2004 [37] sequenced yeast *K. waltii* genome, compared it with yeast *S. cerevisiae* genome, and demonstrated that nearly every region in *K. waltii* corresponds to two regions in *S. cerevisiae* thus proving that there was a whole genome duplication event in the course of yeast evolution. This discovery quickly

followed by the discovery of the whole genome duplications in vertebrates [34, 16] and plants [28]. Recently Dehal and Boore [18] found an evidence of two rounds of whole genome duplications on the evolutionary path from early vertebrates to human. Shortly afterwards, Meyer and Van de Peer [45] found an evidence of yet another (third) round of whole genome duplications in ray-finned fishes.

These recent studies provided an irrefutable evidence that the whole genome duplications represent a new type of events that may explain phenomena that the classical evolutionary studies had difficulties explaining (e.g., emergence of new metabolic pathways [37]). At the same time, they raised a problem of reconstructing the genomic architecture of the ancestral pre-duplicated genomes, named the *Genome Halving Problem*. The Genome Halving Problem was studied in a series of papers by El-Mabrouk and Sankoff [22, 23, 21] culminating in a rather complex algorithm in [24]. The El-Mabrouk–Sankoff algorithm is one of the most technically challenging results in computational biology and its proof spans over 30 pages in [24].

In this dissertation we describe an alternative approach to the Genome Halving Problem based on the notion of *contracted breakpoint graph*. In particular, we identify a flaw in the El-Mabrouk–Sankoff analysis in the case of circular unichromosomal genomes and give a full analysis of the Genome Halving Problem. We remark that our approach is very different from [24] and we do not know whether the technique in [24] can be adjusted to address the described complication.

We further analyze a generalization of Genome Halving Problem for a more general set of rearrangement operations (including transpositions) and propose an efficient algorithm for solving this problem.

1.3 Multi-Break Rearrangements

The "standard" rearrangement operations (i.e., reversals, translocations, fusions, fissions) can be modelled by making 2 breaks in a genome and gluing the resulting fragments in a new order. One can imagine a hypothetical *k-break* rearrangement operation that makes k breaks in a genome and further glues the resulting

pieces in a new order. In particular, the human genome can be modelled as a mouse genome broken into ≈ 280 pieces that are glued together in the "mouse" order.

Most biologists believe that k-break rearrangements are unlikely for $k > 3$ and relatively rare for $k = 3$ (at least in mammalian evolution). Indeed, biophysical limitations and selective constraints are already severe for $k = 2$, let alone for $k > 2$. However, 3-break rearrangements (e.g., transpositions) undoubtedly happen in evolution, although it is still unclear how frequent they are in mammalian evolution. Therefore, it would be useful to generalize the Pevzner–Tesler arguments against RBM as well as the Genome Halving Problem for the case of k-breaks (and 3-breaks in particular). Also, in radiation biology, chromosome aberrations for $k > 2$ (indicative of chromosome damage rather than evolutionary viable variations) may be more common, e.g., complex rearrangements in irradiated human lymphocytes [60, 41, 71, 59].

Thus, the "RBM vs. FBM" controversy, analysis of whole genome duplications, and radiation/cancer biology all call for studies of k-break rearrangements for $k > 2$.

1.4 Dissertation Outline

In Chapter 2 we study multi-break rearrangements in depth. In particular, in Sections 2.1 and 2.2 we show how to compute the *k-break distance* (i.e., the minimum number of k-breaks required to transform one genome into the other) between circular genomes, and in Section 2.3 we derive exact formulas for the k-break distance for small k and give efficient algorithms for computing it in the case of general k. In Section 2.4 we use k-breaks to estimate the breakpoint reuse rate between the human and mouse genomes to support the Pevner–Tesler arguments against the RBM in the presence of complex rearrangements. Finally, in Section 2.5 we extent these results to the harder case of linear genomes.

In Chapter 3 we study the Genome Halving Problem for different types of genomes and rearrangement operations. We start with discussion of the problem of computing the rearrangement distance between genomes with duplicated genes in

Section 3.1. Then we introduce the notion of contracted breakpoint graph in Section 3.2 and show how to use it to solve the Genome Halving Problem in Section 3.3. We solve the 2-Break Genome Halving Problem and the 3-Break Genome Halving Problem for multichromosomal circular genomes in Section 3.4. In Section 3.5 we study the Genome Halving Problem for unichromosomal circular genomes. In particular, we revisit the El-Mabrouk–Sankoff result for unichromosomal circular genomes and describe a flaw in their analysis. We show that this flaw is a rule rather than a pathological case: it affects a large family of duplicated genomes. We further proceed to give a full analysis of the Genome Halving Problem that is based on introducing an invariant that divides the set of all rearranged duplicated genomes into 2 classes. We show that the El-Mabrouk–Sankoff formula is correct for the first class and is off by 1 for the second class.

In Chapter 4 we conclude with a brief summary of contribution and discuss possible directions for future work.

2 Genome Rearrangements

In this chapter we initiate studies of multi-break rearrangements. We prove a duality theorem for the k-break distance between genomes with n genes that shows how to compute it. In particular, we present a dynamic programming algorithm with the running time $O(n^{\lfloor k/2 \rfloor - 2}) + O(n)$ that is practical for small values of k. We also show how one can compute the k-break distance in linear time in n for an arbitrary k that requires preliminary computations that are exponential in k but independent of n. In Section 2.4 we apply these results to the analysis of rearrangements and the "FBM vs. RBM" controversy.

Unless stated otherwise, we deal with *circular genomes* that consist of one or more circular chromosomes. Extension to the harder case of linear genomes is described in Section 2.5.

2.1 Genome Rearrangements and Breakpoint Graphs

From an algorithmic perspective, the genome is a collection of chromosomes, and each chromosome is a sequence of genes. DNA has two strands and genes on a chromosome have directionality that reflects the strand of the genes. We represent the order and directions of the genes on each chromosome as a circular sequence of *signed* elements (i.e., elements with signs "+" and "-"). We distinguish between *unichromosomal genomes* consisting of just a single chromosome and *multichromosomal genomes* consisting of one or more chromosomes.

2.1.1 Unichromosomal Genomes

For unichromosomal genomes, the rearrangements are limited to *reversals* that "flip" genes $x_i \dots x_j$ in a genome $x_1 x_2 \dots x_n$ as follows:

$$x_1 \dots x_{i-1} \underrightarrow{\ x_i \ \ x_{i+1} \dots \ \ x_j\ } x_{j+1} \dots x_n$$

$$\downarrow$$

$$x_1 \dots x_{i-1} \underleftarrow{-x_j - x_{j-1} \cdots - x_i} x_{j+1} \dots x_n$$

The *reversal distance* $d(P, Q)$ between genomes P and Q is defined as the minimal number of reversals required to transform one genome into the other (see Chapter 10 of [55] for a review of genome rearrangement algorithms).

A duality theorem and a polynomial algorithm for computing reversal distance between two signed permutations was proposed by Hannenhalli and Pevzner [30] and later was generalized for multichromosomal genomes [29]. The algorithm was further simplified and improved in a series of papers [6, 35, 1, 7, 68, 36] and applied in a variety of biological studies [46, 12, 10, 54, 5].

We will find it convenient to represent a circular chromosome with genes $x_1, \dots x_n$ as a cycle (Fig. 2.1) composed of n directed labelled edges (corresponding to genes) and n undirected unlabeled edges (connecting adjacent genes). The directions of the edges correspond to *signs* (strand) of the genes. We label the tail and head of a directed edge x_i as x_i^t and x_i^h respectively. Vertex x_i^t is called the *obverse* of vertex x_i^h, and vice versa. Vertices in a chromosome connected by an undirected edge are called *adjacent*.

Let P and Q be circular signed permutations (unichromosomal genomes) on the same set of elements (genes) $\mathcal{G}$. The *breakpoint graph* $G(P, Q)$ is defined on the set of vertices $V = \{x^t, x^h \mid x \in \mathcal{G}\}$ with edges of three colors[1]: "obverse", black, and gray (Fig. 2.1). Edges of each color form a matching on V:

- pairs of obverse elements form an *obverse matching*;

[1]We have chosen rather unusual names for the "colors" (obverse, black, and gray) to be consistent with previous papers on genome rearrangements.

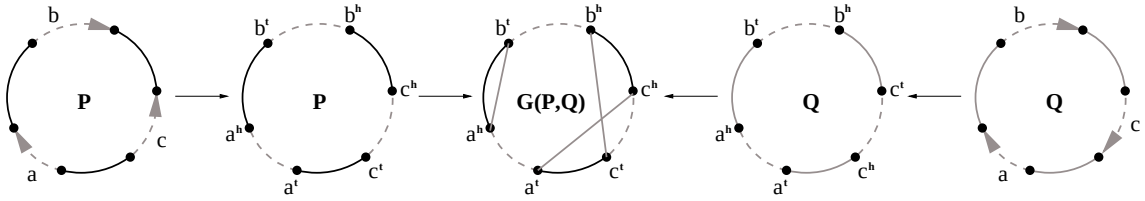

Figure 2.1 The breakpoint graph $G(P, Q)$ of unichromosomal genomes $P = +a+b-c$ and $Q = +a+b+c$ represented as a black-obverse cycle and a gray-obverse cycle correspondingly.

- adjacent elements in P, other than obverses, form a *black matching*;
- adjacent elements in Q, other than obverses, form a *gray matching*.

Every pair of matchings forms a collection of *alternating* cycles in G, called *black-gray*, *black-obverse*, and *gray-obverse* cycles respectively (a cycle is alternating if colors of its edges alternate). The genome P can be read along a single black-obverse cycle while the genome Q can be read along a single gray-obverse cycle in G. The black-gray cycles in the breakpoint graph play an important role in computing the reversal distance. According to the Hannenhalli–Pevzner theorem, the reversal distance between permutations P and Q is given by the formula:

$$d(P, Q) = |P| - c(P, Q) + \mathfrak{h}(P, Q) \tag{2.1}$$

where $|P| = |Q|$ is the size of P and Q, $c(P, Q) = c(G(P, Q))$ is the number of black-gray cycles in the breakpoint graph $G(P, Q)$, and $\mathfrak{h}(P, Q)$ is an easily computable combinatorial parameter (see Chapter 10 of [55] for background information on genome rearrangements).

2.1.2 Multichromosomal genomes

Similarly to unichromosomal genomes, we represent a multichromosomal genome as a collection of disjoint cycles (chromosomes) with edges of two *alternating* colors: one color (black) reserved for undirected edges and the other (obverse) color reserved for directed edges. We do not explicitly show the directions of edges since they are defined by superscripts "t" and "h" (Fig. 2.1).

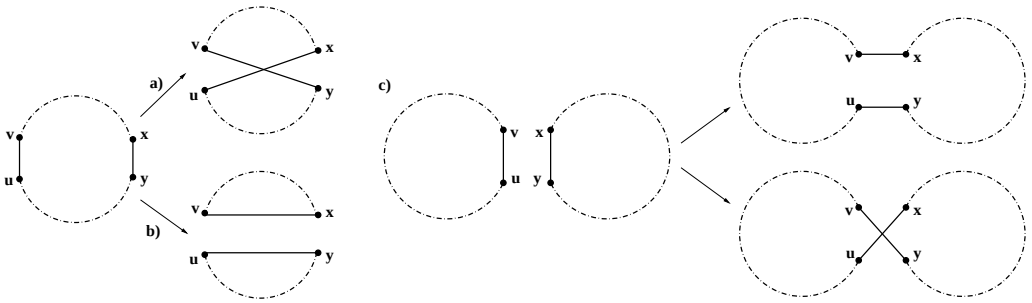

Figure 2.2 2-Breaks on edges (u, v) and (x, y) corresponding to a) Reversal: the edges belong to the same black-obverse cycle that is rearranged after the 2-break; b) Fission: the edges belong to the same black-obverse cycle that is split by the 2-break; c) Translocation/fusion: the edges belong to different black-obverse cycles that are joined by the 2-break.

Let P be a genome represented as a collection of alternating black-obverse cycles (a cycle is alternating if colors of its edges alternate). For any two black edges (u, v) and (x, y) in the genome (graph) P we define a *2-break* rearrangement as replacement of these edges with either a pair of edges (u, x), (v, y), or a pair of edges (u, y), (v, x) (Fig. 2.2). 2-breaks correspond to standard rearrangement operations of reversals (Fig. 2.2a), fissions (Fig. 2.2b), or fusions/translocations[2] (Fig. 2.2c). 2-break rearrangements can be generalized as follows. Given k black edges forming a matching on $2k$ vertices, define a *k-break* as replacement of these edges with a set of k black edges forming another matching in on the same set of $2k$ vertices. Note that a 2-break is a particular case of a 3-break (as well as of a k-break for $k > 3$), in which case only two edges are replaced and the third one remains the same.

While 2-breaks correspond to standard rearrangements, 3-breaks add transposition-like operations (transpositions and inverted transpositions) as well as 3-way fissions to the set of rearrangements (Fig. 2.3). A *transposition* cuts off a continuous segment of one chromosome and inserts it into the same or another chromosome. A transposition of a segment $\pi_i\pi_{i+1}\dots\pi_j$ of a chromosome

$$\pi_1\pi_2\dots\underline{\pi_i\pi_{i+1}\dots\pi_j}\dots\pi_k\pi_{k+1}\dots\pi_m$$

[2]This definition of elementary rearrangement operations follows the standard definitions of reversals, translocations, fissions, and fusions for the case of circular chromosomes. For circular chromosomes fusions and translocations are not distinguishable.

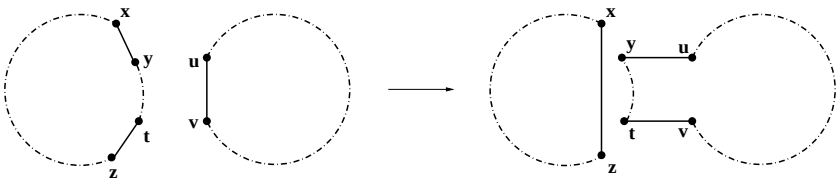

Figure 2.3 A 3-break on edges (u, v), (x, y) and (z, t) corresponding to a transposition of a segment $y \dots t$ from one chromosome to another.

into a position k of the same chromosome results a chromosome

$$\pi_1\pi_2 \dots \pi_{i-1}\pi_{j+1} \dots \pi_k\underline{\pi_i\pi_{i+1} \dots \pi_j}\pi_{k+1} \dots \pi_m.$$

For chromosomes $\pi = \pi_1\pi_2 \dots \pi_i\pi_{i+1} \dots \pi_j \dots \pi_m$ and $\sigma = \sigma_1\sigma_2 \dots \sigma_n$ a transposition of a segment $\pi_i\pi_{i+1} \dots \pi_j$ of chromosome π into a position k in the chromosome σ results in chromosomes

$$\pi_1\pi_2 \dots \pi_{i-1}\pi_{j+1}\pi_{j+2} \dots \pi_m \quad \text{and} \quad \sigma_1\sigma_2 \dots \sigma_{k-1}\underline{\pi_i\pi_{i+1} \dots \pi_j}\sigma_k \dots \sigma_n.$$

Let P and Q be two multichromosomal genomes on the same set of genes $\mathcal{G}$. Similarly to the unichromosomal case, the *breakpoint graph* $G(P, Q)$ is defined on the set of vertices $V = \{x^t, x^h \mid x \in \mathcal{G}\}$ with edges of three colors: "obverse", black, and gray. Edges of each color form a matching on V: *obverse matching* (pairs of obverse vertices), *black matching* (adjacent vertices in P), and *gray matching* (adjacent vertices in Q). Every pair of matchings forms a collection of *alternating* cycles in G, called *black-gray*, *black-obverse*, and *gray-obverse* cycles respectively. The chromosomes of genome P (resp. Q) represent black-obverse (resp. gray-obverse) cycles in $G(P, Q)$.

2.2 Multi-Break Distance Problem

Every k-break in the genome P corresponds to a transformation of the breakpoint graph $G(P, Q)$. Since the breakpoint graph of two identical genomes is a collection of *trivial* black-gray cycles of length 2 (the *identity breakpoint graph*), the problem of transforming the genome P into the genome Q by k-breaks can be formu-

lated as the problem of transforming the breakpoint graph $G(P, Q)$ into the identity breakpoint graph. This is equivalent to the following problem:

k-Break Distance Problem. *Given two perfect matchings (black and gray) in a graph, find a shortest series of k-breaks that transforms one matching into the other.*

In difference from the Genomic Distance Problem [29, 69, 50] (for linear multichromosomal genomes), the 2-Break Distance Problem for circular multichromosomal genomes is trivial (compare to [76]):

Theorem 2.2.1. *The 2-break distance between a black matching P and a gray matching Q is $|P| - c(P, Q)$ where $c(P, Q)$ is the number of black-gray cycles in $G(P, Q)$.*

Proof. It is easy to see that every non-trivial black-gray cycle can be split into two by a 2-break. Since no 2-break can increase the number of black-gray cycles by more than 1, the 2-break distance between P and Q is $|P| - c(P, Q)$. □

In difference from standard rearrangements (modelled as 2-breaks), transpositions introduce 3 breaks in the genome, making them notoriously difficult to analyze. Computing the minimum number of transpositions transforming one genome into another is called *sorting by transpositions*. After Bafna and Pevzner, 1995 [3] gave a first 1.5-approximation algorithm for sorting by transpositions, a number of faster algorithms with the same approximation ratio were proposed [15, 73, 31] culminating in a recent 1.375-approximation algorithm by Elias and Hartman [25]. A number of researchers considered transpositions in conjunction with other rearrangement operations [2, 27, 32, 42, 43, 57, 72]. The complexity of sorting by transpositions remains unknown.

Let $c^{odd}(P, Q)$ be the number of black-gray cycles in the breakpoint graph $G(P, Q)$ with an odd number of black edges (*odd cycles*). The 3-break distance theorem has a simple proof that is very similar to the arguments in [3]:

Theorem 2.2.2. *The 3-break distance between a black matching P and a gray matching Q is $\frac{|P| - c^{odd}(P,Q)}{2}$.*

Proof. A trivial black-gray cycle is a cycle with a single black edge. If $Q = P$, the breakpoint graph $G(P, Q)$ is a set of $|P|$ trivial cycles that are odd cycles (each with a single black edge). It is easy to see that as soon as there is a non-trivial black-gray odd cycle, it can be split into 3 odd cycles by a 3-break, thus increasing the number of odd cycles by 2. On the other hand, if there exists a black-gray even cycle, it can be split into two odd cycles, thus again increasing the number of odd cycles by 2. Since no 3-break can increase the number of black-gray cycles by more than 2, the 3-break distance is $\frac{|P|-c^{odd}(P,Q)}{2}$. □

In Section 3.4.2 we further illustrate the theoretical advantages of considering the 3-break distance (as compared to the transposition distance) by showing that some very difficult problems can be solved if one moves from transpositions to 3-breaks.

Below we prove the duality theorem for the k-break distance for an arbitrary k. A black-gray cycle is called an i_k-*cycle* if it has i modulo $k-1$ black edges. A subset of cycles in a breakpoint graph $G(P, Q)$ is called *breakable* if the total number of black edges in these cycles equals 1 modulo $k-1$. Let $s_k(P, Q)$ be the maximum number of disjoint breakable subsets in $G(P, Q)$. For example, for $k = 3$, every odd cycle forms a breakable subset and $s_3(P, Q) = c^{odd}(P, Q)$. Let $c_k^i(P, Q)$ be the number of black-gray i_k-cycles in $G(P, Q)$. For $k = 4$, every 1_4-cycle forms a breakable subset and every pair of 2_4-cycles forms a breakable subset, implying that $s_4(P, Q) = c_4^1(P, Q) + \lfloor c_4^2(P, Q)/2 \rfloor$. Below we prove that the k-break distance is $d_k(P, Q) = \left\lceil \frac{|P|-s_k(P,Q)}{k-1} \right\rceil$.

We introduce a few definitions. A k-break β and a cycle c are called *compatible* if β either does not use edges of c or uses all its black edges. Otherwise β and c are called *incompatible*. Given a k-break β, we define $\text{def}(\beta)$ as the number of cycles in $G(P, Q)$ that are incompatible with β. Obviously, a k-break β may increase the number of trivial cycles by at most $k - \text{def}(\beta)$. A k-break β is called *optimal* if it is compatible with all cycles in $G(P, Q)$ and if it increases the number of trivial cycles by k. A k-break β with $\text{def}(\beta) = 1$ is called *semi-optimal* if it increases the number

of trivial cycles by $k-1$.

Lemma 2.2.3. *A set S of non-trivial black-gray cycles with m black edges can be transformed into m trivial cycles with $\frac{m-1}{k-1}$ k-breaks if S is breakable and with $\left\lceil \frac{m}{k-1} \right\rceil$ k-breaks otherwise.*

Proof. We first prove that any set S of non-trivial black-gray cycles with m black edges can be transformed into m trivial cycles with a series of $\left\lceil \frac{m}{k-1} \right\rceil$ k-breaks. It is easy to see that if $m > k$ then either an optimal or a semi-optimal k-break exists. Indeed, let $c_1, \ldots, c_t$ be a set of non-trivial cycles in S containing at least k black edges while $c_1, \ldots, c_{t-1}$ contains less than k black edges. If $c_1, \ldots, c_t$ contain exactly k black edges then there exists an optimal k-break using all black edges of these cycles. If $c_1, \ldots, c_t$ contain more than k black edges then there exists a semi-optimal k-break using all black edges of $c_1, \ldots, c_{t-1}$ and some black edges of c_t. In either case, the number of trivial cycles is increasing by at least $k-1$ with every k-break. To complete the proof (for non-breakable sets) it is sufficient to notice that every set of cycles with k or less black edges can be transformed into trivial cycles by a single k-break.

We showed above how to transform a set S into m trivial cycles with a series of optimal and semi-optimal k-breaks (with a possible exception of the last k-break). If one of these k-breaks is optimal, the bound $\left\lceil \frac{m}{k-1} \right\rceil$ turns into $\left\lceil \frac{m-1}{k-1} \right\rceil$ (since each optimal k-break creates k trivial cycles as compared to $k-1$ trivial cycles for semi-optimal k-breaks). It is easy to see that for a breakable set S there exists at least one optimal k-break in the series. □

Theorem 2.2.4. *The k-break distance between a black matching P and a gray matching Q is $\left\lceil \frac{|P|-s_k(P,Q)}{k-1} \right\rceil$.*

Proof. We first prove that there exists a series of $\left\lceil \frac{|P|-s_k(P,Q)}{k-1} \right\rceil$ k-breaks transforming $G(P,Q)$ into a set of trivial cycles. Let $\mathcal{S}$ be a collection of $s_k(P,Q)$ disjoint breakable subsets of black-gray cycles in $G(P,Q)$ and M be the total number of black edges in $\mathcal{S}$. Lemma 2.2.3 implies that every breakable set with m black edges can be decomposed

into trivial cycles with $\frac{m-1}{k-1}$ k-breaks. Therefore, all $s_k(P,Q)$ breakable sets from $\mathcal{S}$ can be decomposed into M trivial cycles with $\frac{M-s_k(P,Q)}{k-1}$ k-breaks. Lemma 2.2.3 also implies that all remaining cycles (i.e., cycles that do not belong to elements of $\mathcal{S}$) with $|P|-M$ black edges in total can be broken into trivial cycles by $\left\lceil \frac{|P|-M}{k-1} \right\rceil$ k-breaks. Therefore, all cycles can be transformed into trivial cycles by $\frac{M-s_k(P,Q)}{k-1} + \left\lceil \frac{|P|-M}{k-1} \right\rceil = \left\lceil \frac{|P|-s_k(P,Q)}{k-1} \right\rceil$ k-breaks.

We now prove that a k-break on $G(P,Q)$ can reduce the value of $\left\lceil \frac{|P|-s_k(P,Q)}{k-1} \right\rceil$ by at most 1, or equivalently, that every k-break can increase $s_k(P,Q)$ by at most $k-1$. Every k-break can create at most k "new" cycles, implying that $s_k(P,Q)$ can increase by at most k. Assume that a k-break β increases $s_k(P,Q)$ by k. Let $\mathcal{S}$ be a maximum set of disjoint breakable subsets of black-gray cycles after performing the k-break β (i.e., $|\mathcal{S}| = s_k(P,Q)+k$). The k-break β may be viewed as a replacement of some "old" cycles $c'_1,\ldots,c'_t$ in $G(P,Q)$ with k "new" cycles $c_1,\ldots,c_k$. Therefore, the total number of black edges in these cycles is the same: $\sum_{i=1}^{k} b(c_i) = \sum_{i=1}^{t} b(c'_i)$ where $b(\cdot)$ denotes the total number of black edges in a subgraph g.

Note that if for each "new" cycle c_i $(i=1,\ldots,k)$ we remove from $\mathcal{S}$ a breakable subset contains c_i, then the remaining breakable subsets will contain only cycles from $G(P,Q)$, implying that the number of remaining subsets cannot exceed $s_k(P,Q)$. Therefore, each "new" cycle c_i $(i=1,\ldots,k)$ must belong to a distinct breakable subset $\mathcal{B}_i \in \mathcal{S}$ with $e_i + b(c_i)$ black edges in total, where $e_i = b(\mathcal{B}_i \setminus \{c_i\})$. Since $e_i + b(c_i)$ equals 1 modulo $k-1$, $\sum_{i=1}^{k} e_i + \sum_{i=1}^{t} b(c'_i) = \sum_{i=1}^{k} e_i + b(c_i)$ equals 1 modulo $k-1$ as well. Therefore, the cycles $c'_1,\ldots,c'_t$ together with the cycles from all $\mathcal{B}_i \setminus \{c_i\}$ form a breakable subset $\mathcal{B}'$. Then the set $(\mathcal{S} \setminus \{\mathcal{B}_1,\ldots,\mathcal{B}_k\}) \cup \{\mathcal{B}'\}$ consists of $s_k(P,Q)+1$ disjoint breakable subsets of black-gray cycles in $G(P,Q)$, a contradiction to the definition of $s_k(P,Q)$. It proves that every k-break can increase $s_k(P,Q)$ by at most $k-1$. □

Theorem 2.2.4 and the formula for $s_4(P,Q)$ imply a formula for the 4-break distance.

Corollary 2.2.5. *The 4-break distance between a black matching P and a gray match-*

ing Q is

$$d_4(P,Q) = \left\lceil \frac{|P| - c_4^1(P,Q) - \lfloor c_4^2(P,Q)/2 \rfloor}{3} \right\rceil.$$

Similarly, one can derive a formula for the 5-break distance, which we state below without a proof.

Corollary 2.2.6. *The 5-break distance between a black matching P and a gray matching Q is*

$$d_5(P,Q) = \left\lceil \frac{|P| - c_5^1(P,Q) - \min\{c_5^2(P,Q), c_5^3(P,Q)\} - \lfloor \max\{0, c_5^3(P,Q) - c_5^2(P,Q)\}/3 \rfloor}{4} \right\rceil.$$

For $k > 5$, a formula for the k-break distance becomes more complicated, e.g., $d_6(P,Q) = \left\lceil \frac{|P| - s_6(P,Q)}{5} \right\rceil$ where

$$\begin{aligned} s_6(P,Q) = c_6^1(P,Q) &+ \left\lfloor \tfrac{c_6^3(P,Q)}{2} \right\rfloor + \min\{c_6^2(P,Q), c_6^4(P,Q)\} \\ &+ \left\lfloor \tfrac{\max\{0, c_6^2(P,Q) - c_6^4(P,Q)\}}{3} \right\rfloor + \left\lfloor \tfrac{\max\{0, c_6^4(P,Q) - c_6^2(P,Q)\}}{4} \right\rfloor + \delta \end{aligned}$$

and δ is either 0 or 1, and $\delta = 1$ iff (i) $c_6^3(P,Q)$ is odd, (ii) $c_6^4(P,Q) > c_6^2(P,Q)$, and (iii) $c_6^4(P,Q) - c_6^2(P,Q)$ equals 2 or 3 modulo 4.

From the algorithmic perspective, while the k-break distance between genomes with n genes can be computed in $O(n)$ time for small k (e.g., for $k \leq 10$), it is unclear whether one can compute $d_k(P,Q)$ in linear time for arbitrary k. In the next Section we address this problem by establishing the relationship between the k-break distance and the Gröbner basis of an appropriately constructed polynomial ideal.

2.3 Algorithms for computing multi-break distance

In this section we present two approaches to computing the k-break distance between genomes with n genes. We start with a dynamic programming algorithm with the running time $O(n^{\lfloor k/2 \rfloor - 2}) + O(n)$ that is practical for small values of k. We further show how one can derive closed-form formulas for the k-break distance via computing the set of so-called extremal breakable vectors. While these formulas lead to linear-time algorithms for a wider range of k, it is not clear how to generalize this approach

for an arbitrary k. Finally, we show how to compute the k-break distance in linear time in n for an arbitrary k (with preliminary computations that are exponential in k but independent of n). While the latter algorithm is linear in theory, the high cost of the preliminary computations makes it less practical than the former algorithms.

2.3.1 Dynamic programming algorithms

First we reformulate the k-break distance as a multi-dimensional packing problem. Since a breakable subset remains breakable after removing all 0_k-cycles, without loss of generality we assume that breakable subsets do not contain 0_k-cycles. Then every breakable subset $\mathcal{B}$ is characterized by a *breakable vector* $v = (v_1, \ldots, v_{k-2})$ where v_i is the number of i_k-cycles in $\mathcal{B}$.

For genomes P and Q, let $c = (c_1, \ldots, c_{k-2})$ where $c_i = c_k^i(P, Q)$. Finding $s_k(P, Q)$ amounts to finding the maximum number of breakable vectors $v^1, \ldots, v^t$ such that $v^1 + \cdots + v^t \leq c$ (component-wise). Note that we can limit our search only to the set V of all *proper* breakable vectors v with $v_j < k - 1$ for all $j = 1, \ldots, k - 2$. Since the first coordinate of a proper breakable vector $v = (v_1, \ldots, v_{k-2})$ is uniquely defined by the others as $v_1 = 1 - 2 \cdot v_2 - \cdots - (k - 2) \cdot v_{k-2} \bmod (k - 1)$, the total number of proper breakable vectors is $|V| = (k - 1)^{k-3}$.

For a vector u with $k - 2$ components, define $s(u)$ as the maximum number of elements of V (each element may appear several times) with the sum not exceeding u. Then $s_k(P, Q) = s(c)$. We will use this formula and Theorem 2.2.4 to come up with an algorithm for computing the k-break distance for an arbitrary k.

Theorem 2.3.1. *For genomes P and Q with n genes, $d_k(P, Q)$ can be computed in $O(n^{k-2}) + O(n)$ time.*

Proof. It is easy to see that $s(u) = \max\limits_{v \in V, v \leq u} s(u - v) + 1$. This formula leads to a dynamic programming algorithm for computing $s_k(P, Q) = s(c)$ via computing $s(u)$ for all $u \leq c$. We need to fill up a dynamic programming table of size $(c_1 + 1) \times \cdots \times (c_{k-2} + 1) = O((n/k)^{k-2})$. Note that the time-complexity of computing each $s(u)$ depends on k but not on n. Therefore, the total time to compute $s_k(P, Q)$

(and $d_k(P,Q)$) is $O(n^{k-2}) + O(n)$, where the term $O(n)$ accounts for time needed to construct the breakpoint graph $G(P,Q)$ and to compute the vector c. □

The following theorem describes a faster version of the dynamic programming approach.

Theorem 2.3.2. *For genomes P and Q with n genes, $d_k(P,Q)$ can be computed in $O(n^{\lfloor k/2 \rfloor - 2}) + O(n)$ time.*

Proof. Let $\mathcal{S}$ be a maximum set of disjoint breakable subsets of black-gray cycles in $G(P,Q)$. An i_k-cycle and a $(k-i)_k$-cycle ($i = 2, \ldots, k-2$) are called *paired* in $\mathcal{S}$ if they form an element of $\mathcal{S}$. We will show how to transform the set $\mathcal{S}$ into a maximum set $\mathcal{S}'$ of disjoint breakable subsets of black-gray cycles in $G(P,Q)$ such that for every $i = 2, \ldots, k-2$, either all i_k-cycles are paired or all $(k-i)_k$-cycles are paired in $\mathcal{S}'$.

Suppose that for some i there is a non-paired i_k-cycle p (belonging to a breakable subset $\mathcal{B}_1$) and a non-paired $(k-i)_k$-cycle q (belonging to a breakable subset $\mathcal{B}_2$) in $\mathcal{S}$. If $\mathcal{B}_1 = \mathcal{B}_2$ then we replace this subset in $\mathcal{S}$ with a breakable subset $\{p, q\}$. If $\mathcal{B}_1 \neq \mathcal{B}_2$ then we replace $\mathcal{B}_1$ and $\mathcal{B}_2$ in $\mathcal{S}$ with breakable subsets $\{p, q\}$ and $(\mathcal{B}_1 \cup \mathcal{B}_2) \setminus \{p, q\}$. Note that this operation transforms $\mathcal{S}$ into a maximum set of disjoint breakable subsets and increases the number of paired cycles. Therefore, after a number of steps we will arrive at a maximum set $\mathcal{S}'$ of disjoint breakable subsets with the required property.

It is easy to see that the number of breakable subsets in $\mathcal{S}'$ formed by an i_k-cycle and a $(k-i)_k$-cycle equals $p_i = \min\{c_k^i(P,Q), c_k^{k-i}(P,Q)\}$ for $i \neq k/2$ and (for k even) $p_{k/2} = \left\lfloor c_k^{k/2}(P,Q)/2 \right\rfloor$. Let $c' = (0, c_k^2(P,Q) - p_2, \ldots, c_k^{k-2}(P,Q) - p_{k-2})$, except that for even k, the $k/2$-th component $c'_{k/2} = c_k^{k/2}(P,Q) - 2p_{k/2} = c_k^{k/2}(P,Q) \bmod 2$. Then

$$s_k(P,Q) = |\mathcal{S}'| = s(c') + c_k^1(P,Q) + \sum_{i=2}^{\lfloor k/2 \rfloor} p_i.$$

Note that at least $\lfloor (k-3)/2 \rfloor$ coordinates of the vector c' are zero while the $k/2$-th coordinate (for even k) is at most 1. Therefore, the dynamic programming table for

computing $s(c')$ in Theorem 2.3.1 is of size $O(n^{\lfloor k/2 \rfloor - 2})$, reducing the overall complexity of the algorithm to $O(n^{\lfloor k/2 \rfloor - 2}) + O(n)$. □

The big-O notation in both dynamic programming algorithms hides a large constant (directly related to the size of the set V) that is exponential in k. Below we describe how one can significantly reduce this constant.

A vector v *dominates* a vector u if $u \leq v$. The vectors that dominate other vectors can be safely removed from the set V to compute the k-break distance more efficiently. This results in a set of *extremal* breakable vectors V'. In the next section we show how the set of *extremal* breakable vectors can be efficiently computed using Hilbert bases, and explore their relation to an explicit formula for $s_k(P, Q)$. While we are unaware of any theoretical bounds on the number of extremal breakable vectors $|V'|$, the numerical results suggest that it is small as compared to the number of proper breakable vectors $|V|$. Replacing the set of proper breakable vectors V with the set of extremal breakable vectors V' in the dynamic programming algorithms reduces the time-complexity in roughly $|V|/|V'|$ times.

Below we show how to compute the set of extremal breakable vectors via computing a certain Hilbert basis. We further use the set of extremal breakable vectors to interpret the problem of computing the k-break distance in terms of algebraic varieties. Then we employ Gröbner bases to come up with an algorithm for computing the k-break distance (for a fixed k) between two genomes with n genes in $O(n)$ time.

2.3.2 Extremal breakable vectors and closed-form formulas for multi-break distance

Consider an embedding $f : V \longrightarrow C$ of the set V of all proper breakable vectors into a cone:

$$C = \{x \in \mathbb{Z}_+^k \mid a \cdot x = 0\}, \quad a = (-1, 1, 2, \ldots, k-3, k-2, -(k-1))$$

such that

$$V \ni (v_1, \ldots, v_{k-2}) \quad \overset{f}{\longmapsto} \quad (1, v_1, \ldots, v_{k-2}, \frac{\sum_{i=1}^{k-2} iv_i - 1}{k-1}) \in C.$$

Let H be a *Hilbert basis* of the cone C, i.e., the minimal set of vectors such that any point in C can be expressed as an integral non-negative linear combination of vectors in H.

Theorem 2.3.3. *The set of extremal breakable vectors is* $f^{-1}(f(V) \cap H)$.

Proof. Let $H' = f(V) \cap H$ and $V' = f^{-1}(H')$. It can be easily verified that H' consists of all vectors in H with the first coordinate equal to 1.

Let $v \in V$ and S be a set of elements of the Hilbert basis H that appear in the expansion of $f(v)$ with positive coefficients. Since the first coordinate of $f(v)$ is 1, S contains exactly one element h from H', and thus $f^{-1}(h) \leq v$. If v is an extremal vector then $f^{-1}(h) = v$, implying that $f^{-1}(H')$ contains all extremal vectors of V. On the other hand, if v is not extremal then $f(v) \neq h$, implying that the set of all extremal breakable vectors is $f^{-1}(H')$. □

We have computed the Hilbert basis H of the cone C (for $k \leq 20$) using the algorithm from [51], and applied Theorem 2.3.3 to obtain a set of extremal breakable vectors V'. The size of H and V' is listed in Table 2.1.

For small k, the terms in the formula for $s_k(P, Q)$ can be mapped to the set of extremal breakable vectors V'. For example, for $k = 6$, the set of extremal breakable vectors is

$$V' = \{(1,0,0,0), (0,0,2,0), (0,1,0,1), (0,0,1,2), (0,3,0,0), (0,0,0,4)\}$$

Table 2.1 The size of the set V of all proper breakable vectors, of the Hilbert basis H of the cone C, of the set V' of extremal vectors, and of the reduced Gröbner basis GB.

k	$\lvert V\rvert = (k-1)^{k-3}$	$\lvert H\rvert$	$\lvert V'\rvert$	$\lvert GB\rvert$
3	1	3	1	1
4	3	7	2	3
5	16	13	3	9
6	125	27	6	43
7	1296	39	8	125
8	16807	83	16	1117
9	262144	117	22	8227
10	4782969	205	37	
11	100000000	291	53	
12	2357947691	555	92	
13	61917364224	634	110	
14	1792160394037	1277	201	
15	56693912375296	1567	260	
16	1946195068359375	2368	376	
17	72057594037927936	3315	519	
18	2862423051509815793	5740	831	
19	121439531096594251776	6228	963	
20	548038685778480218593 9	11404	1592	

and it is mapped to the terms of the formula for $s_6(P,Q)$ as follows[3]

$$\begin{array}{ccc} (1,0,0,0) & (0,0,2,0) & (0,1,0,1) \\ \downarrow & \downarrow & \downarrow \\ c_6^1(P,Q) & \left\lfloor \frac{c_6^3(P,Q)}{2} \right\rfloor & \min\{c_6^2(P,Q), c_6^4(P,Q)\} \end{array}$$

$$\begin{array}{ccc} (0,0,1,2) & (0,3,0,0) & (0,0,0,4) \\ \downarrow & \downarrow & \downarrow \\ \delta & \left\lfloor \frac{\max\{0, c_6^2(P,Q) - c_6^4(P,Q)\}}{3} \right\rfloor & \left\lfloor \frac{\max\{0, c_6^4(P,Q) - c_6^2(P,Q)\}}{4} \right\rfloor \end{array}$$

This may give a hope for a "simple" formula for $s_k(P,Q)$ that would allow one to compute $d_k(P,Q)$ efficiently. While we indeed were able to achieve it for $k < 10$ (via the Hilbert basis approach), the complexity of such formulas grows very fast with

[3]While knowing V' provides an intuition and facilitates the proof of the formulas for k-break distance, we are not aware of an algorithm to automatically translate V' into a formula for k-break distance.

k (e.g., see how the term "δ" in the formula for $s_6(P,Q)$ is defined).

2.3.3 Computing multi-break distance in linear time

For a field $\mathcal{K}$, consider a polynomial ring $\mathcal{P} = \mathcal{K}[x, y_1, \ldots, y_{k-2}, z_1, \ldots, z_m]$ where $m = |V'|$ is the number of extremal breakable vectors[4]. Let I be an ideal of $\mathcal{P}$ generated by binomials $xy_1^{v_1^i} \ldots y_{k-2}^{v_{k-2}^i} - z_i$, $i = 1, \ldots, m$ where $v^1, \ldots, v^m$ are the elements of V'. Let GB be a reduced *Gröbner basis* of the ideal I w.r.t. the degree of x and the graded reverse lexicographical ordering of the variables $y_1, \ldots, y_{k-2}, z_1, \ldots, z_m$. The following theorem shows how to compute $s(c)$ in constant time using the Gröbner basis GB.

Theorem 2.3.4. *Let N be an integer such that $s(c) \leq N$ (e.g., $N = \sum_{i=1}^{k-2} c_i$), $f = x^N y_1^{c_1} \ldots y_{k-2}^{c_{k-2}}$ be a polynomial in $\mathcal{P}$, and f' be a normal form of f with respect to the Gröbner basis GB. Then $f' = x^{N-s(c)} y_1^{d_1} \ldots y_{k-2}^{d_{k-2}} z_1^{e_1} \ldots z_m^{e_m}$ where $d_1, \ldots, d_{k-2}$, $e_1, \ldots, e_m$ are some non-negative integers. Moreover, $e_1 + \cdots + e_m = s(c)$ and the multiset of vectors $\{(v^1)^{e_1}, \ldots, (v^m)^{e_m}\}$ from V' is of the maximum cardinality with the sum of elements not exceeding c.*

Proof. It follows from the Buchberger algorithm (see [17] for background information on Gröbner bases) that the reduced Gröbner basis of an ideal generated by binomials consists of binomials. Hence, the normal form of the monomial f is a monomial. Suppose that $f' = x^{N'} y_1^{d_1} \ldots y_{k-2}^{d_{k-2}} z_1^{e_1} \ldots z_m^{e_m}$ where $N', d_1, \ldots, d_{k-2}, e_1, \ldots, e_m$ are some non-negative integers.

The definition of the function $s(\cdot)$ implies that there exist non-negative integers $t_1, \ldots, t_m$ such such that $t_1 \cdot v^1 + \cdots + t_m \cdot v^m \leq c$ and $t_1 + \cdots + t_m = s(c)$. Then the polynomial $x^{N-s(c)} y_1^{u_1} \ldots y_{k-2}^{u_{k-2}} z_1^{t_1} \ldots z_m^{t_m}$ belongs to $f + I$ where $u = c - t_1 \cdot v^1 - \cdots - t_m \cdot v^m$.

Since GB is a Gröbner basis of the ideal I, the polynomial f' is minimal in $f + I$. Hence, $N' \leq N - s(c)$. On the other hand, it is easy to see that $e_1 \cdot v^1 +$

[4]We note that the running time of computing a Gröbner basis is highly sensitive to the number of variables. Hence, using the set of extremal breakable vectors V' instead of the set of proper breakable vectors V dramatically reduces the complexity of the Gröbner basis computing.

$\cdots + e_m \cdot v^m \leq c$ and, thus $N - N' = e_1 + \cdots + e_m \leq s(c)$ by the definition of $s(\cdot)$. Therefore, $s(c) = N - N'$. □

For a given k, computing the reduced Gröbner basis GB may take time exponential in k. But as soon as GB is found, computing the k-break distance between genomes P and Q with n genes takes time linear in n. In particular, it takes linear time in n to construct the breakpoint graph $G(P, Q)$ and the vector c to obtain the polynomial f. Then it takes constant time (depending on k) w.r.t. n to compute a normal form of f w.r.t. GB and to obtain the distance between P and Q. For k up to 9, we have computed the reduced Gröbner basis GB using computer algebra system SINGULAR version 3.0.2 [26] (see Table 2.1).

2.4 Computing the Breakpoint Reuse Rate

One of the arguments against the Pevzner–Tesler rebuttal of RBM [56] was recently raised by Sankoff, 2006 [62]:

> ... we cannot infer whether mutually randomized synteny block orderings derived from two divergent genomes were created ... through processes other than reversals and translocations.

We consider this argument for the human genome H and the mouse genome M based on the 281 synteny blocks from [54], assuming that all chromosomes are circular. While analyzing linear chromosomes would be more adequate than analyzing their circularized versions, it poses additional algorithmic challenges that will be addressed in Section 2.5. We will show that that switching to circular chromosomes does not lead to significant changes as compared to linear chromosomes.

The breakpoint graph $G(H, M)$ contains 35 black-gray cycles including 3 odd black-gray cycles, implying that $d_2(H, M) = 281 - 35 = 246$ (Theorem 2.2.1) and $d_3(H, M) = 139$ (Theorem 2.2.2). If each of 139 3-breaks on a shortest evolutionary path from H to M made 3 breaks, it would imply that there were $139 \cdot 3 - 281 = 136$ breakpoint re-uses (for this particular evolutionary path), resulting in the *breakpoint*

re-use rate 1.48 (see Peng et al., 2006 [52]). While this is a high breakpoint re-use rate (inconsistent with RBM and the scan statistics), this estimate relies on the assumption that each 3-break on the evolutionary path from H to M makes 3 breaks (*complete* 3-breaks). In reality, some 3-breaks can make 2 breaks (*incomplete* 3-breaks) as 2-breaks are particular cases of 3-breaks, reducing the estimate for the number of breakpoint re-uses. Moreover, the minimum number of breakpoint re-uses may be achieved on a suboptimal evolutionary path from H to M.

The rebuttal of RBM raises a question about finding a transformation of H into M by 3-breaks that makes the minimal number of individual breaks. The following theorem shows that there exists a series of $d_3(P, Q)$ 3-breaks that makes the minimum number of breaks while transforming P into Q:

Theorem 2.4.1. *Any series of m k-breaks transforming a circular genome P into a circular genome Q makes at least $m + d_2(P, Q)$ breaks. Moreover, there exists a series of $d_3(P, Q)$ 3-breaks transforming P into Q that makes $d_3(P, Q) + d_2(P, Q)$ breaks.*

Proof. For each k-break operation, let $\Delta(cycles)$ be the increase in the number of cycles and $\Delta(breaks)$ be the increase in the number of breaks. It is easy to see that $\Delta(cycles) \leq \Delta(breaks) - 1$. Summing up over a series of m k-breaks transforming P into Q, we have $|P| - c(P, Q) \leq b - m$, where b is the total number of breaks made in the series. Therefore, $b \geq |P| - c(P, Q) + m = d_2(P, Q) + m$.

Consider a shortest series of complete 3-breaks transforming every odd black-gray cycles into trivial cycles and every even black-gray cycle into trivial cycles and a single cycle with two black edges. This series consists of $d_3(P, Q) - c^{even}(P, Q)$ 3-breaks and results in $c^{even}(P, Q)$ cycles with two black edges that can be transformed into trivial cycles with a series of $c^{even}(P, Q)$ incomplete 3-breaks (i.e., 2-breaks). The total number of 3-breaks in this transformation is $d_3(P, Q)$ and they make $3(d_3(P, Q) - c^{even}(P, Q)) + 2c^{even}(P, Q) = 3d_3(P, Q) - c^{even}(P, Q) = d_3(P, Q) + d_2(P, Q)$ breaks overall. □

Corollary 2.4.2. *Every transformation between the circularized human genome H*

and mouse genome M by 3-breaks requires at least 104 breakpoint re-uses (implying that there exist rearrangement hotspots in the human genome).

Proof. Any transformation of H into M requires at least $d_3(H, M) + d_2(H, M) = 139 + 246 = 385$ breaks. Since there are 281 breakpoints between the human and mouse genomes, it implies that there were at least $385 - 281 = 104$ breakpoint re-uses on the evolutionary path from human to mouse, resulting in breakpoint re-use rate 1.37. This is still higher that the expected breakpoint re-use rate of RBM as computed by scan statistics [56]. It provides an argument against RBM not only for $k = 2$ but for $k = 3$ as well and invalidates arguments from [62] in the case $k = 3$ (see also [Ale07]). Since k-breaks for $k > 3$ were never reported in previous evolutionary studies, it is unlikely that they significantly affect our conclusions. □

Theorem 2.4.1 implies that any transformation of the human genome H into the mouse genome M with 2-breaks makes at least $d_2(H, M) + d_2(H, M) = 246 + 246 = 492$ breaks, while any transformation of H into M with 3-breaks makes at least $d_3(H, M) + d_2(H, M) = 139 + 246 = 385$ breaks. Below we show how the lower bound on the number of breaks made in a series of 3-breaks depends on the number of complete 3-breaks in this series.

Theorem 2.4.3. *For any series of m 3-breaks with t complete 3-breaks, transforming a genome P into a genome Q,*

$$m \geq \max\{d_2(P, Q) - t, d_3(P, Q)\}.$$

Moreover, there exists a series of $\max\{d_2(P, Q) - t, d_3(P, Q)\}$ 3-breaks with at most t complete 3-breaks, transforming P into Q.

Proof. Since k-break can increase the number of cycles in the breakpoint graph by at most $k - 1$, a series with t complete 3-breaks and $m - t$ incomplete 3-breaks (i.e., 2-breaks) can increase the number of cycles by at most $2t + (m - t) = m + t$. If it transforms the genome P into the genome Q then $m + t \geq |P| - c(P, Q) = d_2(P, Q)$. Therefore, $m \geq d_2(P, Q) - t$.

Consider a series of complete 3-breaks, transforming every black-gray cycle with $q \geq 3$ black edges into two trivial cycles and a cycle with $q-2$ black edges. Note that such a series may have at most $d_3(P,Q) - c^{even}(P,Q)$ (the longest possible series results in $c^{even}(P,Q)$ cycles with 2 black edges and $|P| - c^{even}(P,Q)$ trivial cycles). Since every such 3-break increases the number of cycles by 2, a series of $q = \min\{t, d_3(P,Q) - c^{even}(P,Q)\}$ such 3-breaks result in $c(P,Q)+2q$ cycles. These cycles can be transformed into trivial cycles with a series of $|P|-(c(P,Q)+2q) = d_2(P,Q)-2q$ 2-breaks. The total number of 3-breaks and 2-breaks in this transformation is

$$q + d_2(P,Q) - 2q = d_2(P,Q) - \min\{t, d_3(P,Q) - c^{even}(P,Q)\} = \max\{d_2(P,Q) - t, d_3(P,Q)\}.$$

□

Theorems 2.4.1 and 2.4.3 imply:

Corollary 2.4.4. *Any series of 3-breaks with t complete 3-breaks, transforming a genome P into a genome Q, makes at least $d_2(P,Q) + \max\{d_2(P,Q) - t, d_3(P,Q)\}$ breaks. In particular, any such series of 3-breaks with $t \leq d_2(P,Q) - d_3(P,Q)$ complete 3-breaks makes at least $2d_2(P,Q) - t$ breaks.*

Corollary 2.4.4 gives the lower bound for the breakpoint re-use rate as a function of the number of complete 3-breaks (i.e., transpositions and 3-way fusions/fissions) in a series of 3-breaks transforming one genome into the other. For the human genome H and mouse genome M, this lower bound is shown in Fig. 2.4a.

Corollaries 2.4.2 and 2.4.4 address only the case of circularized chromosomes and further analysis is needed to extend it to the case of linear chromosomes (see Section 2.5). Recently, Bergeron et al., 2006 [8] described another promising approach to analyzing both circular and linear chromosomes (using DCJ operations proposed in [76]) that also opens a possibility to obtain the breakpoint re-use estimates for linear genomes. However, the above estimate is based on the extreme assumption that certain 3-breaks (transpositions and 3-way fissions) represent the dominant rearrangements while reversals and translocations are extremely rare (contrary to the existing view). We emphasize that we do not share the point of view that genomes

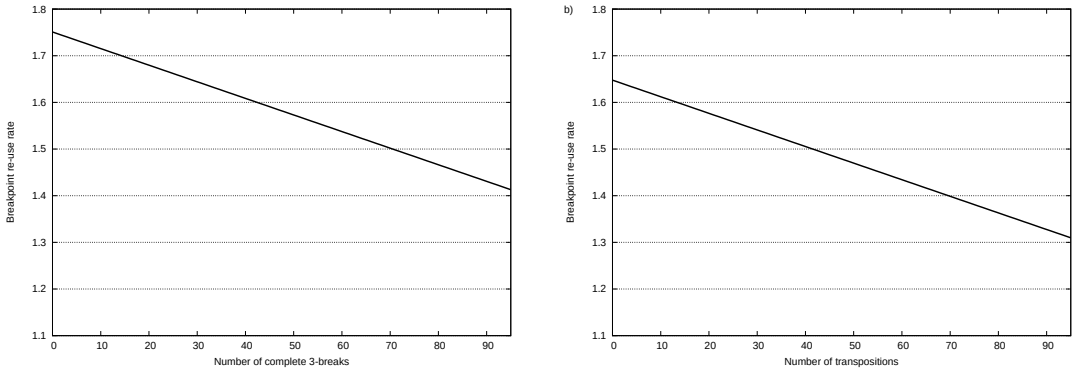

Figure 2.4 The lower bound on the breakpoint re-use rate between the human and mouse genomes based on 281 synteny blocks from [54]. The lower bound is represented as a function of a) the number of complete 3-breaks in a series of 3-breaks between the circularized human and mouse genomes. b) the number of transpositions in a series of rearrangements between the linear human and mouse genomes.

mainly evolve by transpositions and 3-way fissions, and that we analyzed this assumption only to refute the arguments against FBM. A more realistic analysis of 3-breaks leads to a much higher estimate of the breakpoint re-use (see Fig. 2.4).

2.5 Multi-Break Rearrangements and Linear Genomes

While multi-breaks in linear genomes can be defined similarly to circular genomes, the linear case is harder to analyze. In contrast to circular genomes, not every multi-break can be performed over a linear genome: multi-breaks that create circular chromosomes are not allowed. In this section we extend the results from Sections 2.2 and 2.4 to the case of linear genomes.

A *linear genome* is a collection of linear chromosomes represented as sequences of signed elements (genes). Similarly to circular genomes, we represent each linear chromosome on n genes as a sequence of n directed obverse edges (encoding genes and their direction) and $n-1$ undirected black edges (connecting adjacent genes). So, each linear chromosome is an alternating *path* of obverse and black edges (starting and ending with obverse edges), and a linear genome is a collection of such paths.

Every linear genome P with m chromosomes has $2m$ vertices representing endpoints of the chromosomes. If we introduce an arbitrary perfect matching on these $2m$ vertices, consisting of black *closing edges*, the resulting graph will represent some circular genome that contains P as a subgraph. We call the resulting genome a *closure* of P and note that in general it is not uniquely defined. Black edges that belong to P are called *non-closing*.

Throughout this section we assume that P and Q are linear genomes on the same set of genes.

2.5.1 Rearrangement distance between linear genomes

Let $d_2^l(P, Q)$ be the genomic distance between the genomes P and Q, i.e., the minimum number of reversals, translocations, fissions, and fusions required to transform P into Q. Also, let $d_3^l(P, Q)$ be the minimum number of reversals, translocations, fissions, and fusions as well as transpositions[5] required to transform P into Q.

Theorem 2.5.1. *For any closure P' of a genome P, there exists a closure Q' of a genome Q such that $d_2^l(P, Q) \geq d_2(P', Q')$. Similarly, for any closure P' of a genome P, there exists a closure Q' of a genome Q such that $d_3^l(P, Q) \geq d_3(P', Q')$.*

Proof. Let S' be a closure of a linear genome S. We note that any reversal, translocation, fission, or fusions transforming the genome S into a linear genome T corresponds to a 2-break transforming the closure S' into some closure T' of the genome T (Fig. 2.5a,b,c,d). Similarly, any transposition transforming the genome S into a linear genome T corresponds to a 3-break transforming the closure S' into some closure T' of the genome T (Fig. 2.5e).

For the genomes P and Q, consider a series of $d_k^l(P, Q)$ ($k = 2$ or $k = 3$) rearrangements transforming P into Q. This series corresponds to a series of k-breaks transforming P' into some circular genome Q' that is a closure of the genome Q. To

[5]We do not consider 3-way fusions and 3-way fissions since such operations were never reported in biological literature.

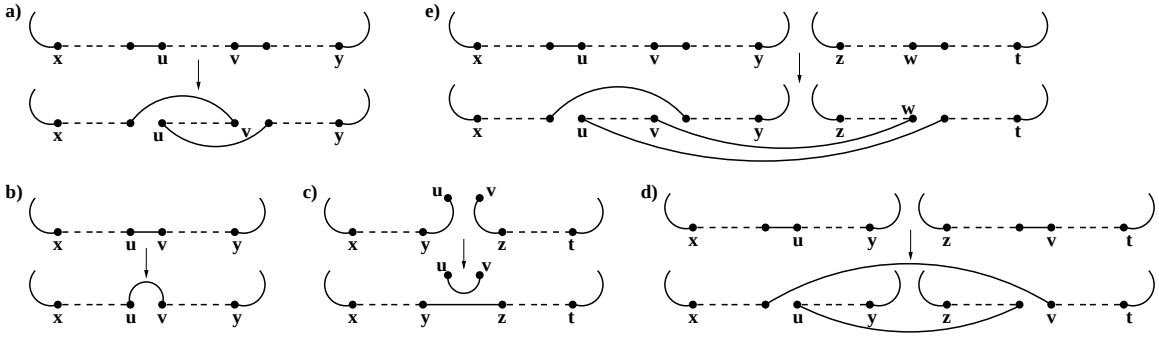

Figure 2.5 Rearrangements of linear genomes correspond to k-breaks over closures: a) Reversal of the region (u, v) is a 2-break over non-closing black edges; b) Fission at the black edge (u, v) is the identity multi-break over the edge (u, v), re-claiming this edge as closing; c) Fusion of the chromosomes endpoints y and z is a 2-break replacing closing edges (y, u) and (z, v) with a non-closing edge (y, z) and a closing edge (u, v); d) Translocation exchanging chromosomes parts (u, y) and (v, t) is a 2-break operating over non-closing edges; e) Transposition is a 3-break operating over non-closing edges.

complete the proof it is sufficient to notice that the distance $d_k(P', Q')$ between the genomes P' and Q' does not exceed $d_k^l(P, Q)$, i.e., $d_k(P', Q') \leq d_k^l(P, Q)$. □

Theorem 2.5.1 immediately implies:

Corollary 2.5.2. *For any linear genomes P and Q, $k = 2$ or $k = 3$,*

$$d_k^l(P, Q) \geq \max_{P'} \min_{Q'} d_k(P', Q')$$
$$d_k^l(P, Q) = d_k^l(Q, P) \geq \max_{Q'} \min_{P'} d_k(P', Q').$$

where P' and Q' vary over all possible closures of the genomes P and Q respectively.

Since the k-break distance $d_k(P', Q')$ ($k = 2$ or $k = 3$) gives a lower bound for the linear distance $d_k^l(P, Q)$, our goal is to make this bound as tight as possible by choosing appropriate closures P' and Q'. We start with defining the breakpoint graph of linear genomes and a number of its characteristics that we will find useful.

Let P' and Q' be closures of linear genomes P and Q. The breakpoint graph $G(P, Q)$ is defined as a result of removal of all closing edges from the breakpoint graph $G(P', Q')$ (of circular genomes P' and Q'). It is easy to see that $G(P, Q)$ is well-defined by the genomes P and Q and does not depend on a particular choice of

closures P' and Q'. Every cycle in $G(P', Q')$ with m closing edges will be split into m paths in $G(P, Q)$. Therefore, the black-gray connected components of $G(P, Q)$ are formed by $c(P, Q)$ black-gray cycles and a number of black-gray paths. We distinguish between black-gray paths with both terminal edges of black color (*bb-paths*), with both terminal edges of gray color (*gg-paths*), and with terminal edges of different colors (*bg-paths*), including isolated vertices viewed as bg-paths with zero black and zero gray edges. We denote the number of such paths by $l_{bb}(P, Q)$, $l_{gg}(P, Q)$, and $l_{bg}(P, Q)$ respectively (note that the number $l_{bg}(P, Q)$ is always even). The total number of black-gray connected components in $G(P, Q)$ is

$$cc(P, Q) = c(P, Q) + l_{bb}(P, Q) + l_{gg}(P, Q) + l_{bg}(P, Q).$$

We also distinguish between black-gray connected components with odd/even number of black/gray edges and call them *b-odd*, *b-even*, *g-odd*, *g-even* respectively. To refer to the number of such components we will use these aliases as superscripts. Similarly to cycles, bg-paths have the same number of black and gray edges, so we call bg-paths simply *odd* and *even*, depending on the oddness of the number of black edges. We denote the number of such bg-paths by $l_{bg}^{odd}(P, Q)$ and $l_{bg}^{even}(P, Q)$ respectively. We rely on the following identities:

$$\begin{aligned}
&\forall j \in \{bb, bg, gg\},\\
&l_j(P, Q) = l_j^{b-odd}(P, Q) + l_j^{b-even}(P, Q), \quad l_j(P, Q) = l_j^{g-odd}(P, Q) + l_j^{g-even}(P, Q);\\
&\forall j \in \{bb, gg\},\\
&l_j^{b-odd}(P, Q) = l_j^{g-even}(P, Q), \qquad\qquad l_j^{b-even}(P, Q) = l_j^{g-odd}(P, Q).
\end{aligned}$$

These identities allow to compute all the parameters as soon as $c(P, Q)$, $c^{odd}(P, Q)$, $l_{bb}(P, Q)$, $l_{bb}^{b-odd}(P, Q)$, $l_{gg}(P, Q)$, $l_{gg}^{b-odd}(P, Q)$, $l_{bg}(P, Q)$, $l_{bg}^{odd}(P, Q)$ are given.

Similarly to the breakpoints graphs for circular and linear genomes, we can define breakpoint graphs and associated characteristics in the case when one genome is circular while the other is linear (in such a graph all paths are either bb-paths or gg-paths).

Lemma 2.5.3. *For a circular genome P' and a linear genome Q,*

$$\min_{Q'} d_2(P', Q') = |P'| - cc(P', Q) \quad \textit{and} \quad \min_{Q'} d_3(P', Q') = \frac{|P'| - cc^{b-odd}(P', Q)}{2}$$

where Q' varies over all possible closures of the genome Q.

Proof. Theorem 2.2.1 implies that $\min_{Q'} d_2(P', Q') = |P'| - \max_{Q'} c(P', Q')$. To maximize $c(P', Q')$, the closure Q' needs to be chosen in such a way that it closes each path in the breakpoint graph $G(P', Q)$ into a separate black-gray cycle. Therefore, $\max_{Q'} c(P', Q') = cc(P', Q)$.

Similarly, Theorem 2.2.2 implies that

$$\min_{Q'} d_3(P', Q') = \frac{|P'| - \max_{Q'} c^{odd}(P', Q')}{2}.$$

To maximize $c^{odd}(P', Q')$, the closure Q' needs to be chosen in such a way that it closes each b-odd path in the breakpoint graph $G(P', Q)$ into a separate black-gray cycle. Therefore, $\max_{Q'} c(P', Q') = cc^{b-odd}(P', Q)$. □

Theorem 2.5.4. *For linear genomes P and Q, $\max_{P'} \min_{Q'} d_2(P', Q') = B_2(P, Q)$ where*

$$B_2(P, Q) = |P| - c(P, Q) - \max\{1, \frac{l_{bg}(P, Q)}{2}\} - l_{bb}(P, Q),$$

implying that $d_2^l(P, Q) \geq \max\{B_2(P, Q), B_2(Q, P)\}$.

Proof. By Lemma 2.5.3 we have $\max_{P'} \min_{Q'} d_2(P', Q') = |P| - \min_{P'} cc(P', Q)$. In order to minimize $cc(P', Q)$, the closure P' needs to be chosen in such a way that it minimizes the number of black-gray connected components in $G(P, Q)$. This can be done as follows. If $l_{bg}(P, Q) = 0$, then we will connect (using closing black edges) all the gg-paths into a single cycle. If $l_{bg}(P, Q) > 0$, we will first connect a pair of bg-paths and all the gg-paths into a single bb-path, and then form pairs of the remaining bg-paths and connect bg-paths in each pair into a bb-path. As a result, $\min_{P'} cc(P', Q) = c(P, Q) + \max\{1, \frac{l_{bg}(P,Q)}{2}\} + l_{bb}(P, Q)$. Therefore, $\max_{P'} \min_{Q'} d_2(P', Q') = B_2(P, Q)$ and by Corollary 2.5.2, $d_2^l(P, Q) \geq B_2(P, Q)$. Moreover, since $d_2^l(P, Q) = d_2^l(Q, P) \geq B_2(Q, P)$, we have $d_2^l(P, Q) \geq \max\{B_2(P, Q), B_2(Q, P)\}$. □

Lemma 2.5.5. *For linear genomes P and Q,* $\min_{P'} cc^{b-odd}(P', Q) = L_3(P, Q)$ *where*

$$\begin{aligned} L_3(P,Q) &= c^{odd}(P,Q) + l_{bb}^{b-odd}(P,Q) + \delta(P,Q) \\ &+ \max\left\{0, \frac{|l_{bg}^{odd}(P,Q) - l_{bg}^{even}(P,Q)|}{2} - l_{gg}^{b-even}(P,Q)\right\} \end{aligned}$$

and $\delta(P,Q) = \max\left\{0, l_{gg}^{b-even}(P,Q) - \frac{|l_{bg}^{odd}(P,Q) - l_{bg}^{even}(P,Q)|}{2}\right\} \bmod 2$.

Proof. Note that in any closure of P, the closing (black) edges connect gg-paths and bg-paths from $G(P,Q)$ into $m_1 = \frac{l_{bg}(P,Q)}{2}$ bb-paths and a number of cycles. Note that if $l_{bg}(P,Q) = 0$ then connecting all gg-paths into a single cycle (which will be odd iff $l_{gg}^{b-even}(P,Q)$ is odd) gives an optimal closure P'' (i.e., for which $\min_{P'} cc^{b-odd}(P', Q) = cc^{b-odd}(P'', Q)$). It is easy to check that in this case $cc^{b-odd}(P'', Q) = L_3(P, Q)$. For the rest of the proof we assume that $l_{bg}(P,Q) > 0$.

We will show that there exists an optimal closure where the closing edges do not connect any gg-paths into a cycle. Such an optimal closure can be obtained from an arbitrary optimal closure P'' as explained below. Since $l_{bg}(P,Q) > 0$, the closing edges in $G(P'', Q)$ create at least one bb-path formed by two bg-paths at the ends and possibly gg-paths in the middle. Let us re-connect (modifying the set of closing edges) all the gg-paths from $G(P,Q)$, that are connected into cycles in $G(P'', Q)$, in the middle of this bb-path. Note that such modification of the closure may change the b-oddness of the affected bb-path but only if at least one of the destroyed cycles was odd. In any case the number of b-odd connected components is not increased. Therefore, the modified closure is optimal and satisfies the required property by construction. Without loss of generality we will assume that the closing edges create no cycles.

Bringing black closing edges into $G(P,Q)$ can be viewed as a two-step process: first, connecting gg-paths into longer gg-paths; and second, connecting pairs of bg-paths and maybe single gg-paths into bb-paths. Our goal is to minimize the number of b-odd bb-paths or, equivalently, to maximize the number of b-even bb-paths.

Consider an outcome of the first step. It is clear that connection of two b-odd gg-paths or two b-even gg-paths results in a b-odd gg-path, while connection

of b-odd and b-even gg-paths results in b-even gg-path. As we will see b-even gg-paths are more preferable than b-odd gg-paths. After the first step we can have up to $m_2 = l_{gg}^{b-even}(P,Q)$ b-even gg-paths.

Now, consider the second step. Connection of an odd bg-path and an even bg-path with an optional b-odd gg-path in between create a b-even bb-path. At the same time connection of a pair of odd bg-paths or a pair of even bg-paths requires a b-even gg-path in between in order to produce a b-even bb-path. All other combinations of bg-paths and gg-paths result in b-odd bb-paths.

We can create $m_3 = \min\{l_{bg}^{odd}(P,Q), l_{bg}^{even}(P,Q)\}$ b-even bb-paths without any use of gg-paths, and up to $m_4 = \frac{|l_{bg}^{odd}(P,Q) - l_{bg}^{even}(P,Q)|}{2}$ b-even bb-paths (note that $m_3 + m_4 = m_1$), each of which requires a b-even gg-path in the middle. Hence, we can create $m_5 = m_3 + \min\{m_4, m_2\} = \min\{m_1, m_2 + m_3\}$ b-even bb-paths. The other $m_6 = m_1 - m_5 = \max\{0, m_4 - m_2\}$ bb-paths (formed by pairs of bg-paths of the same oddness) will be b-odd. So far we have used $\min\{m_4, m_2\}$ b-even gg-paths. The other gg-paths (if any) can be connected (at the first step) into a single gg-path that is b-odd iff $m_2 - \min\{m_4, m_2\} = \max\{m_2 - m_4, 0\}$ is odd (i.e., $\delta(P,Q) = 1$). The b-odd gg-path can be easily incorporated into any of created bb-paths without changing its b-oddness. The b-even gg-path we have to incorporate into some of created b-even bb-paths and turn it into a b-odd bb-path. Hence, for an optimal closure P', there are $l_{bb}^{b-odd}(P,Q) + m_6 + \delta(P,Q)$ b-odd bb-paths and $c^{odd}(P,Q)$ odd cycles in $G(P',Q)$, implying that $cc^{b-odd}(P',Q) = c^{odd}(P,Q) + l_{bb}^{b-odd}(P,Q) + m_6 + \delta(P,Q)$. □

Theorem 2.5.6. *For linear genomes P and Q, $d_3^l(P,Q) \geq \max\{B_3(P,Q), B_3(Q,P)\}$ where $B_3(P,Q) = \frac{|P| - L_3(P,Q)}{2}$.*

Proof. Since $d_3^l(P,Q) = d_3^l(Q,P)$ it is sufficient to show that $d_3^l(P,Q) \geq B_3(P,Q)$. Corollary 2.5.2 and Lemma 2.5.3 imply

$$d_3^l(P,Q) \geq \max_{P'} \min_{Q'} d_3(P',Q') = \frac{|P| - \min_{P'} cc^{b-odd}(P',Q)}{2}.$$

Now, applying Lemma 2.5.5 completes the proof. □

2.5.2 Breakpoint re-use in linear genomes

Similarly to the case of circular genomes, we are interested in estimating the total number breaks required to transform a linear genome P into a linear genome Q with reversals, fusions, fissions, translocations, and transpositions. According to Theorem 2.5.1, any series of such rearrangements corresponds to a series of 3-breaks transforming some closure P' of the genome P into some closure Q' of the genome Q. Let $b^c(P, Q)$ be the minimum number of breaks made in such a series of 3-breaks (over all possible closures P' and Q'). Theorems 2.5.1 and 2.4.1 imply:

Corollary 2.5.7. *For linear genomes P and Q,*

$$b^c(P, Q) \geq \max_{P'} \min_{Q'} d_3(P', Q') + d_2(P', Q')$$
$$b^c(P, Q) = b^c(Q, P) \geq \max_{Q'} \min_{P'} d_3(P', Q') + d_2(P', Q')$$

where P' and Q' vary over all possible closures of the genomes P and Q respectively.

To find out the exact value of $\max_{P'} \min_{Q'} d_3(P', Q') + d_2(P', Q')$ we need the following lemma:

Lemma 2.5.8. *For a circular genome P' and a linear genome Q,*

$$\min_{Q'} d_2(P', Q') + d_3(P', Q') = \frac{3}{2}|P'| - \frac{3cc^{b-odd}(P', Q) + 2cc^{b-even}(P', Q)}{2}.$$

Proof. Theorems 2.2.1 and 2.2.2 imply that

$$\min_{Q'} d_3(P', Q') + d_2(P', Q') = \frac{3}{2}|P'| - \frac{\max_{Q'} 3c^{odd}(P', Q') + 2c^{even}(P', Q')}{2}.$$

To maximize $3c^{odd}(P', Q') + 2c^{even}(P', Q')$, a closure Q' has to be chosen in such a way that it closes each path in the breakpoint graph $G(P', Q)$, into a separate black-gray cycle. Indeed, having $m > 1$ paths connected into a single cycle is always worse than connecting each of these paths into a separate cycle as $3 < 2m$. Therefore, for an optimal closure Q', we have $c^{odd}(P', Q') = cc^{b-odd}(P', Q)$ and $c^{even}(P', Q') = cc^{b-even}(P', Q)$. □

Theorem 2.5.9. *For linear genomes P and Q,* $\max_{P'} \min_{Q'} d_3(P', Q') + d_2(P', Q') = B_{23}(P, Q)$ *where*

$$B_{23}(P, Q) = \frac{3}{2}|P| - c(P, Q) - l_{bb}(P, Q) - \frac{l_{bg}(P, Q) + L_3(P, Q)}{2},$$

implying that $b^c(P, Q) \geq \max\{B_{23}(P, Q), B_{23}(Q, P)\}$.

Proof. By Lemma 2.5.8 we have

$$\max_{P'} \min_{Q'} d_3(P', Q') + d_2(P', Q') = \frac{3}{2}|P| - \frac{\min_{P'} 3cc^{b-odd}(P', Q) + 2cc^{b-even}(P', Q)}{2}.$$

Note that if $l_{bg}(P, Q) = 0$ then connecting all gg-paths into a single cycle (which will be odd iff $l_{gg}^{b-even}(P, Q)$ is odd) gives an optimal closure P'. It is easy to check that in this case $\max_{P'} \min_{Q'} d_3(P', Q') + d_2(P', Q') = B_{23}(P, Q)$. For the rest of the proof we assume that $l_{bg}(P, Q) > 0$.

Note that in any closure of P, the closing (black) edges connect gg-paths and bg-paths from $G(P, Q)$ into bb-paths and cycles. We will show that in an optimal closure the closing edges do not connect any gg-paths into a cycle. Indeed, since $l_{bg}(P, Q) > 0$, the closing edges create at least one bb-path formed by two bg-paths at the ends and possibly gg-paths in the middle. It is easy to see that it is always better to include more gg-paths in the middle of this bb-path (maybe letting the objective function increase by one) rather than to create a separate cycle out of these gg-paths (in which case the objective function would increase by at least 2). Therefore, closing edges in an optimal closure P' connect gg-paths and bg-paths from $G(P, Q)$ into $\frac{l_{bg}(P,Q)}{2}$ bb-paths in $G(P', Q)$. As the total number of new bb-paths is fixed, the problem of minimizing $3cc^{b-odd}(P', Q) + 2cc^{b-even}(P', Q)$ is equivalent to minimizing $cc^{b-odd}(P', Q)$. For an optimal closure P', Lemma 2.5.5 gives $cc^{b-odd}(P', Q) = L_3(P, Q)$, implying that

$$\begin{aligned} 3cc^{b-odd}(P', Q) + 2cc^{b-even}(P', Q) &= 2cc(P', Q) + cc^{b-odd}(P', Q) \\ &= 2(c(P, Q) + l_{bb}(P, Q) + \tfrac{l_{bg}(P,Q)}{2}) + L_3(P, Q) \end{aligned}$$

and thus $\max_{P'} \min_{Q'} d_3(P', Q') + d_2(P', Q') = B_{23}(P, Q)$.

By Corollary 2.5.7 we have $b^c(P, Q) \geq B_{23}(P, Q)$ and $b^c(P, Q) \geq B_{23}(Q, P)$, implying that $b^c(P, Q) \geq \max\{B_{23}(P, Q), B_{23}(Q, P)\}$. □

We will now prove the following analog of Corollary 2.4.4:

Theorem 2.5.10. *Any series of rearrangements with t transpositions, transforming a linear genome P into a linear genome Q, makes at least* $\max\{2B_2(P,Q) - t, B_{23}(P,Q)\} - \mathrm{chr}(P) + \mathrm{chr}(Q)$ *breaks, where* $\mathrm{chr}(\cdot)$ *denotes the number of chromosomes. In particular, any such series of rearrangements with* $t \leq 2B_2(P,Q) - B_{23}(P,Q)$ *transpositions makes at least* $2B_2(P,Q) - \mathrm{chr}(P) + \mathrm{chr}(Q) - t$ *breaks.*

Proof. Any series of rearrangements transforming the genome P into the genome Q corresponds to series of 3-breaks transforming any particular closure P' into a closure Q' (depending on P'). We note that every rearrangement makes the same number of breaks as the corresponding 3-break in the closures;[6] except for fusions that make smaller number of breaks than the corresponding 2-breaks in the closure (Fig. 2.5b), and for fissions that make breaks in linear genomes but correspond to identity multi-breaks (making no breaks) over the closures (Fig. 2.5c).

Let u, v, t be respectively the number of fusions, fissions, and transpositions in a series of m rearrangements transforming the genome P into the genome Q and making b breaks in total. Then there is a series of 3-breaks, transforming a closure P' into a closure Q', that makes $b + u - v$ breaks in total. Since every fusion decreases the number of chromosomes by one, while every fission increases the number of chromosomes by one, $u - v = \mathrm{chr}(P) - \mathrm{chr}(Q)$. By Theorem 2.4.3,

$$b + u - v = b + \mathrm{chr}(P) - \mathrm{chr}(Q) \geq d_2(P', Q') + \max\{d_2(P', Q') - t, d_3(P', Q')\},$$

implying that $b \geq \max\{2d_2(P', Q') - t, d_2(P', Q') + d_3(P', Q')\} + \mathrm{chr}(Q) - \mathrm{chr}(P)$. Taking $\max_{P'} \min_{Q'}$ of the right hand side of this inequality, we have $b \geq \max\{2B_2(P,Q) - t, B_{23}(P,Q)\} + \mathrm{chr}(Q) - \mathrm{chr}(P)$. □

Using 281 synteny blocks between the linear human genome H and mouse

[6] We assume that a transposition always makes 3 breaks even if it transposes a part of chromosome starting with one of its ends, a translocation always makes 2 breaks even if it exchanges an entire chromosome with a part of another chromosome, and a reversal always makes 2 breaks even if it involves an end of a chromosome. The biological rationale for this assumption is that chromosomes are flanked by telomeres that while remaining "invisible" in genomic sequences, can account for breakpoint re-use in the same way as any other genomic position.

genome M from [54], we estimate the breakpoint re-use rate across these (linear) genomes. The breakpoint graph $G(H, M)$ have the following parameters: $(c, l_{bb}, l_{gg}, l_{bg}) = (28, 12, 15, 16)$, $(c^{odd}, l_{bb}^{b-odd}, l_{gg}^{b-odd}, l_{bg}^{odd}) = (2, 5, 4, 3)$, $\mathrm{chr}(H) = 23$, $\mathrm{chr}(M) = 20$, $B_2(H, M) = 233$, $B_2(M, H) = 230$, $B_3(H, M) = 137$, $B_3(M, H) = 134$, $B_{23}(H, M) = 370$, and $B_{23}(M, H) = 364$. Theorems 2.5.4 and 2.5.6 imply that

$$d_2^l(H, M) \geq \max\{B_2(H, M), B_2(M, H)\} = 233,$$
$$d_3^l(H, M) \geq \max\{B_3(H, M), B_3(M, H)\} = 137.$$

Theorem 2.5.10 gives the lower bound for the breakpoint re-use rate between the genomes H and M, shown in Fig. 2.4b (as the function of the number of transpositions). This illustrates that very large number of transpositions would be necessary to bring the breakpoint re-use rate below 1.25 rate expected for RBM (see [56]). Therefore, Sankoff's argument that high breakpoint re-use rate reported for human-mouse genomic architectures is an artifact caused by not accounting for complex rearrangements [62] may only hold if one assumes that transpositions are dominant rearrangement operations that are more frequent than reversals, translocations, fissions, and fusions. While detailed analysis of such an extreme rearrangement scenario remains beyond the scope of our analysis we remark that currently there is no biological evidence to support this scenario.

Acknowledgements

This chapter is based on the following three papers:

- Max A. Alekseyev and Pavel A. Pevzner. "Multi-Break Rearrangements and Chromosomal Evolution". *Theoretical Computer Science*, 2007. (to appear)
- Max A. Alekseyev and Pavel A. Pevzner. "Are There Rearrangement Hotspots in the Human Genome?". *PLoS Computational Biology*, 2007. (to appear)
- Max A. Alekseyev. "Multi-Break Rearrangements: from Linear to Circular Genomes". *Proceedings of the 5th Annual RECOMB Satellite Workshop on Comparative Genomics*, 2007. (to appear)

The dissertation author was the primary investigator and author of these papers.

3 Whole Genome Duplications and Genome Halving Problem

The *whole genome duplication* doubles the gene content of a genome R and results in a *perfect duplicated genome* Q that contains two copies of each chromosome of R. The genome then becomes subject to rearrangements that shuffle the genes in Q resulting in some *duplicated genome* P. The *Genome Halving Problem* is to reconstruct the ancestral perfect duplicated genome Q from the given duplicated genome P (Fig. 3.1a).

We represent a circular chromosome R as a cycle formed by directed edges encoding the genes and their direction (Fig. 3.1b, center). There are two natural ways to represent duplication of the chromosome R resulting in a single chromosome $R \oplus R$ (Fig. 3.1b, left) or in two chromosomes $2R$ (Fig. 3.1b, right) but only the former one is applicable to unichromosomal genomes. A *unichromosomal duplicated genome* is a result of a series of reversals applied to the *unichromosomal perfect duplicated genome* $R \oplus R$. The Genome Halving Problems for unichromosomal genomes is formulated as follows:

Genome Halving Problem (unichromosomal genomes). *Given a unichromosomal duplicated genome P, find a perfect unichromosomal duplicated genome $R \oplus R$ minimizing the reversal distance $d(P, R \oplus R)$.*

A whole genome duplication of a multichromosomal genome consisting of chromosomes $R_1, \ldots, R_k$ results in a *multichromosomal perfect duplicated genome*[1]

[1]Note that in difference from the unichromosomal genomes, the whole genome duplication of a multi-

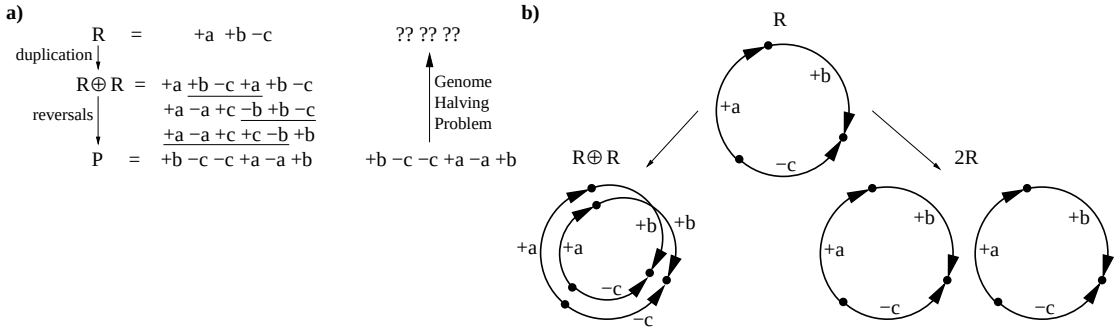

Figure 3.1 a) Whole genome duplication of a genome $R = +a + b - c$ into a perfect duplicated genome $R \oplus R = +a + b - c + a + b - c$ followed by three reversals. b) Whole genome duplication of a circular chromosome R (center) resulting in $R \oplus R$ (left) or $2R$ (right).

where every chromosome R_i is duplicated either into $R_i \oplus R_i$ or into $2R_i$ (Fig. 3.1b). A *multichromosomal duplicated genome* is a result of a series of 2-breaks applied to a perfect duplicated genome. The Genome Halving Problem for multichromosomal genomes (2-Break Genome Halving Problem) is formulated as follows.

Genome Halving Problem (multichromosomal genomes). *Given a duplicated genome P, find a perfect duplicated genome Q minimizing the genomic distance $d_2(P, Q)$.*

The Genome Halving Problem was studied in a series of papers by El-Mabrouk and Sankoff [22, 23, 21] culminating in a rather complex algorithm in [24]. The El-Mabrouk–Sankoff algorithm is one of the most technically challenging results in computational biology and its proof spans over 30 pages in [24]. In this chapter we revisit the El-Mabrouk–Sankoff work and present an alternative approach for the case of unichromosomal genomes. The crux of our approach is a new construction that generalizes the notion of breakpoint graph for any set of genomes with duplicated genes (any gene may be present in an arbitrary number of copies). This construction is related to well-known de Bruijn graphs and proved to be useful in studies of the Genome Halving Problem.

Our studies of the contracted breakpoint graph led us to realize that El-

chromosomal genome is not uniquely defined.

Mabrouk–Sankoff analysis has a flaw and the problem of finding $\min_R d(P, R \oplus R)$ remains unsolved for unichromosomal genomes. Below we show that this flaw is a rule rather than a pathological case: it affects a large family of duplicated genomes. We further proceed to give a full analysis of the Genome Halving Problem that is based on introducing an invariant that divides the set of all rearranged duplicated genomes into 2 classes. We show that the El-Mabrouk–Sankoff formula is correct for the first class and is off by 1 for the second class. We remark that our approach is very different from [24] and we do not know whether the technique in [24] can be adjusted to address the described complication.

We also solve a novel 3-Break Genome Halving Problem for multichromosomal genomes (which includes transpositions into the set of rearrangements operations):

3-Break Genome Halving Problem. *Given a duplicated genome P, find a perfect duplicated genome Q minimizing the 3-break distance $d_3(P, Q)$.*

This chapter is organized as follows. Section 3.1 discusses the problem of computing rearrangement distance between duplicated genomes and formulates the Weak Genome Halving Problem (for unichromosomal genomes). Section 3.2 presents the concept of contracted breakpoint graph for the case of multichromosomal genomes. We solve the Genome Halving Problem for multichromosomal genomes and the 3-Break Genome Halving Problem in Sections 3.4.1 and 3.4.2 respectively. Section 3.5.1 describes a flaw in El-Mabrouk-Sankoff analysis. Section 3.5.2 classifies the genomes for which the original El-Mabrouk–Sankoff theorem is incorrect. Finally, Section 3.5.3 presents our Genome Halving Algorithm for unichromosomal genomes.

3.1 Rearrangement Distance Between Duplicated Genomes

While the Hannenhalli–Pevzner theory leads to a fast algorithm for computing reversal distance between two signed permutations, the problem of computing reversal distance between two genomes with duplicated genes remains unsolved.

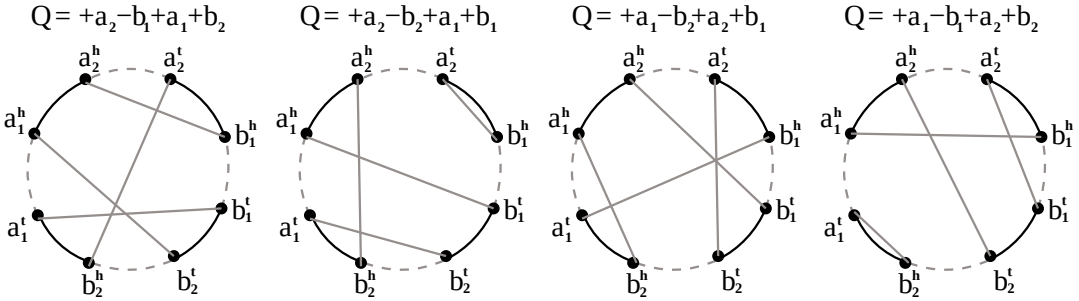

Figure 3.2 Breakpoint graphs corresponding to four different labellings of genomes $P = +a - a - b + b$ and $Q = +a - b + a + b$ (without loss of generality we assume that labelling of $P = +a_1 - a_2 - b_1 + b_2$ is fixed). Two out of four breakpoint graphs have $c(G) = 1$, while two others have $c(G) = 2$.

Let P and Q be duplicated genomes on the same set of genes $\mathcal{G}$ (i.e., each gene appears in two copies). If one labels copies of each gene x as x_1 and x_2 then genomes P and Q become signed permutations and the Hannenhalli–Pevzner theory applies. As before we turn the labelled genomes P and Q into unsigned permutations $\pi(P)$ and $\pi(Q)$ by replacing each element x_i with a pair of obverses $x_i^t x_i^h$ in the order defined by the sign of x_i. Breakpoint graph $G(P, Q)$ of the labelled genomes P and Q has a vertex set $V = \{x_1^t, x_1^h, x_2^t, x_2^h \mid x \in \mathcal{G}\}$ and uniquely defines permutations $\pi(P)$ and $\pi(Q)$ (and, thus, the original genomes P and Q) as well as an inter-genome correspondence between gene copies.

We remark that different labellings may lead to different breakpoint graphs for the same genomes P and Q (Fig. 3.2) and it is not clear how to choose a labelling that results in the minimum reversal distance between the labelled copies of P and Q.

Recently there were many attempts to generalize the Hannenhalli-Pevzner theory for genomes with duplicated and deleted genes [11, 14, 20, 61, 66, 67]. However, the only known option for solving the reversal distance problem for duplicated genomes *exactly* is to consider all possible labellings, to compute the reversal distance problem for each labelling, and to choose the labelling with the minimal reversal distance. For duplicated genomes with n genes this leads to 2^n invocations of the Hannenhalli–Pevzner algorithm rendering this approach impractical. Moreover,

the problem remains open if one of the genomes is perfectly duplicated (i.e., computing the reversal distance $d(P, R \oplus R)$). Surprisingly, the problem of computing $\min_R d(P, R \oplus R)$ that we address in this paper is solvable in polynomial time.

Using the concept of the breakpoint graph and formula (2.1), the Genome Halving Problem can be posed as follows. For a given duplicated genome P, find a perfect duplicated genome $R \oplus R$ and a labelling of gene copies such that the breakpoint graph $G(P, R \oplus R)$ of the labelled genomes P and $R \oplus R$ attains the minimum value of $|P| - c(G) + \mathfrak{h}(G)$. Since $|P|$ is constant and the existing results [13] suggests that $\mathfrak{h}(G)$ is typically small, the value of $d(P, Q)$ depends mostly on $c(G)$. El-Mabrouk and Sankoff [24] established that the problems of maximizing $c(G)$ and minimizing $\mathfrak{h}(G)$ can be solved separately in a consecutive manner. In this dissertation we focus on the former and harder problem:

Weak Genome Halving Problem. *For a given duplicated genome P, find a perfect duplicated genome $R \oplus R$ and a labelling of gene copies that maximizes the number of black-gray cycles $c(G)$ in the breakpoint graph $G(P, R \oplus R)$ of the labelled genomes P and $R \oplus R$.*

For multichromosomal genomes, while Theorem 2.2.1 leads to a polynomial algorithm for computing the 2-break distance between genomes with non-duplicated genes, it is unclear how one can compute this distance between duplicated genomes without going over all possible labellings of the genomes. In the next section we describe the contracted breakpoint graphs that address this complication.

3.2 Contracted Breakpoint Graphs and Labelling Problem

To introduce breakpoint graphs of genomes with duplicated genes we first revisit the notion of breakpoint graph and discuss the relationships between breakpoint graphs and de Bruijn graphs. We find it convenient to represent a circular signed permutation as an alternating cycle formed by edges of two colors with one color reserved for directed obverse edges. For example, Fig. 3.3a,b shows a black-

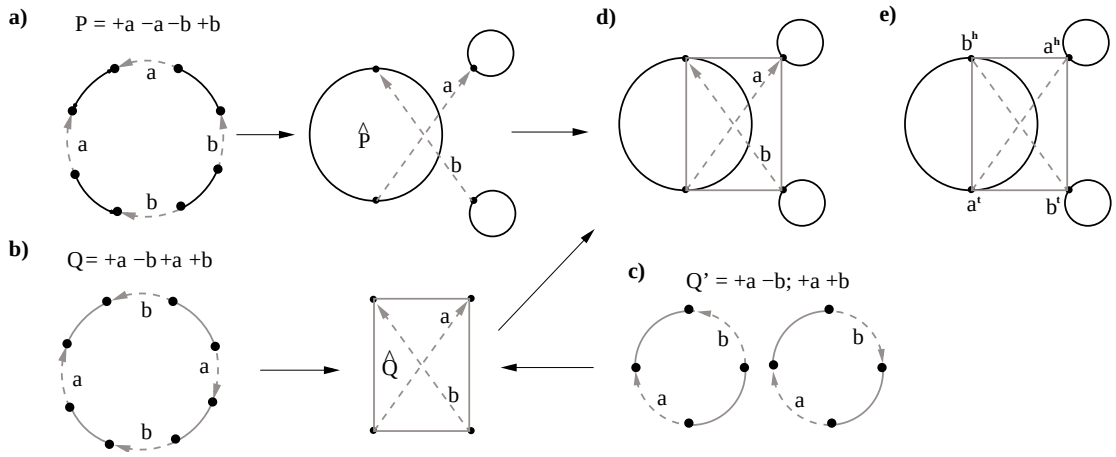

Figure 3.3 a) Genome $P = +a-a-b+b$ as a black-obverse cycle and its transformation into $\hat{P}$ by gluing identically labelled edges; b) Genome $Q = +a-b+a+b$ as a gray-obverse cycle and its transformation into $\hat{Q}$ by gluing identically labelled edges; c) Two-chromosomal genome $Q' = (+a-b)(+a+b)$ that is equivalent to the genome Q ($\hat{Q'} = \hat{Q}$); d) de Bruijn graph for P and Q; e) Contracted breakpoint graph $G'(P, Q)$.

obverse cycle representation of permutation $P = +a - a - b + b$ and a gray-obverse cycle representation of permutation $Q = +a - b + a + b$ (the obverse edges in these cycles are labelled). Given a set of edge-labelled graphs, the *de Bruijn graph* of this set is defined as the result of "gluing"[2] edges with the same label in all graphs in the set (compare with Pevzner et al. 2004 [53]). The de Bruijn graph for two cycles in Fig. 3.3a,b is shown in Fig. 3.3d.

For any genome P (represented as a cycle) we define $\hat{P}$ as the graph obtained from P by gluing identically labelled edges. Obviously, the de Bruijn graph of P and Q coincides with the de Bruijn graph of $\hat{P}$ and $\hat{Q}$ (Fig. 3.3).

While our definition of the de Bruijn graphs is somewhat different from the usual definition, one can see that it produces the same graphs. For example, the classical de Bruijn graph $G_l(x_1 \dots x_n)$ of a circular sequence $x_1 \dots x_n$ parameterized with an integer $l \geq 2$ is defined as a graph with vertices corresponding to all $(l-1)$-tuples and edges corresponding to all l-tuples that occur in $x_1 \dots x_n$ (edge $x_i \dots x_{i+l-1}$ connects vertices $x_i \dots x_{i+l-2}$ and $x_{i+1} \dots x_{i+l-1}$). One can see that $G_l(x_1 \dots x_n)$ is

[2]Gluing takes into the directions of edges, i.e., tails (or heads) of all edges with a given label are glued into a single vertex.

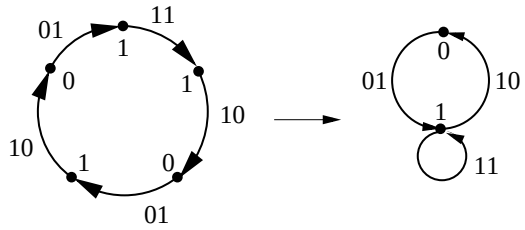

Figure 3.4 The de Bruijn graph $G_2(01101)$ of the circular sequence 01101.

identical to our construction if the circular sequence $x_1 \dots x_n$ is first represented as a cycle passing through all l-tuples in the sequence with further gluing of identically labelled edges of this cycle (Fig. 3.4).

While our de Bruijn graph construction is merely an equivalent definition of the breakpoint graph it provides an important new insight. While it was not clear how to generalize the classical notion of breakpoint graph for genomes with duplicated genes, the de Bruijn graphs automatically provide such a generalization. In fact, the de Bruijn graphs are defined as a gluing operation on an *arbitrary* set of graphs and therefore are applicable to any set of genomes including multichromosomal ones (each genome is represented as a set of cycles). The contracted breakpoint graph defined below is simply the de Bruijn graph of duplicated genomes.

The conventional breakpoint graph (see Section 2.1.1) of signed permutations P and Q on n elements can be defined as the gluing of n *pairs* of obverse edges in the corresponding permutations $\pi(P)$ and $\pi(Q)$ represented as black-obverse and gray-obverse alternating cycles. The *contracted breakpoint graph* of duplicated genomes P and Q on n elements is simply the gluing of n *quartets* of obverse edges. Below we give an equivalent and a somewhat more formal definition of the contracted breakpoint graph.

Let P and Q be duplicated genomes on the same set of genes $\mathcal{G}$ and G be a breakpoint graph defined by some labelling of P and Q. The *contracted breakpoint graph* $G'(P,Q)$ is the result of contracting every pair of vertices x_1^j, x_2^j (where $x \in \mathcal{G},\ j \in \{t,h\}$) in the breakpoint graph G into a single vertex x^j. So the contracted breakpoint graph $G' = G'(P,Q)$ is a graph on the set of vertices $V' = \{x^t, x^h \mid x \in \mathcal{G}\}$

with each vertex incident to two black, two gray, and a *pair* of parallel obverse edges (Fig. 3.3e). The contracted breakpoint graph $G'(P, Q)$ is uniquely defined by P and Q and does not depend on a particular labelling. The following theorem gives a characterization of the contracted breakpoint graphs for multichromosomal genomes.

Theorem 3.2.1. *A graph H with black, gray, and obverse edges is a contracted breakpoint graph for some duplicated genomes iff each vertex in H is incident to two black edges, two gray edges, and two parallel obverse edges.*

Proof. If H is a contracted breakpoint graph of some duplicated genomes then the theorem follows from the definition of contracted breakpoint graph.

Let H be a graph with each vertex incident to two black edges, two gray edges, and a pair of parallel obverse edges. Label endpoints of each obverse edge x in H by x^t and x^h. Since the black degree of each vertex of H is even and so is obverse degree, there exist alternating black-obverse cycles traversing all black and obverse edges in this graph. These cycles define some duplicated genome P. Similarly, since the gray degree of each vertex of H is even, there exist alternating gray-obverse cycles traversing all gray and obverse edges. These cycles define some duplicated genome Q. Then the graph H is a contracted breakpoint graph for the genomes P and Q. □

In the case when Q is a perfect duplicated genome, the gray edges in the contracted breakpoint graph $G'(P, Q)$ form pairs of parallel gray edges that we refer to as *double* gray edges. Similar to the double obverse edges, the double gray edges form a matching in G' (Fig. 3.6a).

Let $G(P, Q)$ be a breakpoint graph for some labelling of P and Q. A set of black-gray cycles in $G(P, Q)$ is contracted into a set of black-gray cycles in the contracted breakpoint graph $G'(P, Q)$, thus forming a black-gray cycle decomposition of $G'(P, Q)$. Therefore, each labelling of P and Q *induces* a black-gray cycle decomposition of $G'(P, Q)$. We are interested in the following problem:

Labelling Problem. *Given a black-gray cycle decomposition of the contracted breakpoint graph $G'(P, Q)$ of duplicated genomes P and Q, find a labelling of P and Q that*

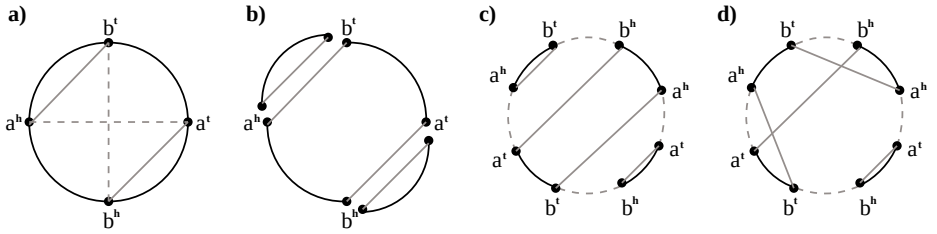

Figure 3.5 a) Contracted breakpoint graph $G'(P, R \oplus R)$ for $P = +a + b - a - b$ and $R = +a + b$; b) Black-gray cycle decomposition C of G' which is not induced by any labelling of P and $R \oplus R$; c) Breakpoint graph $G(P, 2R)$ inducing C; d) Breakpoint graph $G(P, R \oplus R)$ (unique up to re-labelling of vertices) with $c(G) = 2 < |C|$.

induces this cycle decomposition.

This problem may not always have a solution for unichromosomal genomes (Fig. 3.5) and this is exactly the factor that leads to a counterexample to the El-Mabrouk–Sankoff theorem in Section 3.5.1. For multichromosomal genomes, the Labelling Problem can be addressed by considering equivalent genomes.

We call genomes Q and Q' *equivalent* if their de Bruijn graphs are equal, i.e., $\hat{Q} = \hat{Q}'$. If Q and Q' are equivalent then the contracted breakpoint graphs $G'(P, Q)$ and $G'(P, Q')$ are the same for any genome P (Fig. 3.3d).

Lemma 3.2.2. *If Q is a perfect duplicated genome and a genome Q' is equivalent to Q then Q' is perfect duplicated as well.*

Proof. Consider gray and obverse matchings in the de Bruijn graph $\hat{Q} = \hat{Q}'$ formed by pairs of double gray and double obverse edges. These matchings form a set of gray-obverse cycles (consisting of double edges). Every such cycle c is the result of gluing some gray-obverse cycles $c_1, c_2, \ldots, c_k$ in Q' such that $|c_1| + |c_2| + \ldots + |c_k| = 2 \cdot |c|$. Neither of these cycles can be shorter than c since such a short cycle would remain short after gluing. This implies that $k = 1$ or $k = 2$, i.e., the genome Q' has either a single cycle (a chromosome $R \oplus R$) traversing the cycle c two times or two cycles (a pair of chromosomes $2R$) each traversing c once. Therefore, the genome Q' represents a set of sub-genomes of the form $R \oplus R$ or $2R$ implying that Q' is a perfect duplicated genome. □

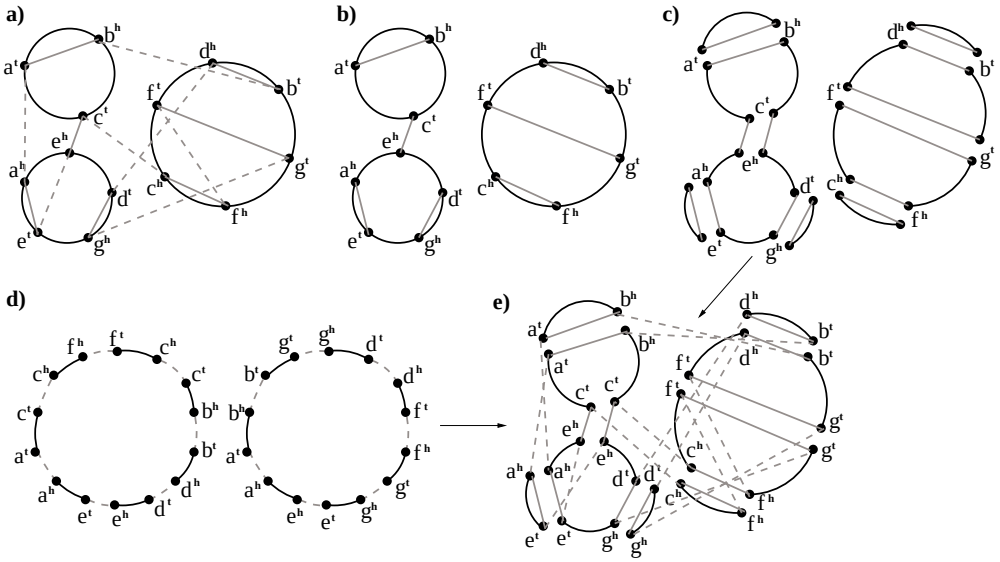

Figure 3.6 For genomes $P = (-a-b+g+d+f+g+e)(-a+c-f-c-b-d-e)$ and $R = -a-b-d-g+f-c-e$, a) contracted breakpoint graph $G'(P, R \oplus R)$; b) BG-graph corresponding to G'; c) maximal black-gray cycle decomposition (split decomposition) C of G' forming graph H; d) genome P as black-obverse cycles; e) breakpoint graph $G(P, Q')$ inducing the cycle decomposition C.

Theorem 3.2.1 and Lemma 3.2.2 imply

Theorem 3.2.3. *A graph H with black, gray, and obverse edges is a contracted breakpoint graph $G'(P, Q)$ for some duplicated genome P and perfect duplicated genome Q iff each vertex in H is incident to two black edges, a double gray edge, and a double obverse edge.*

While the Labelling Problem may not have a solution, the following theorem provides a "compromise" substitute for its solution.

Theorem 3.2.4. *Let P and Q be multichromosomal duplicated genomes and C be a black-gray cycle decomposition of the contracted breakpoint graph $G'(P, Q)$. Then there exists a genome Q' equivalent to Q and a labelling of P and Q' such that the breakpoint graph $G(P, Q')$ induces the cycle decomposition C.*

Proof. Consider a contracted breakpoint graph $G' = G'(P, Q)$ of the genomes P and Q and its black-gray cycle decomposition C (Fig. 3.6a gives an example of a contracted breakpoint graph while Fig. 3.6c gives an example of its black-gray cycle

decomposition). Without loss of generality we can assume that the labelling of P is fixed. In order to prove the theorem we need to find a breakpoint graph $G(P, Q')$ of the labelled genomes P and Q' (Q' is equivalent to Q) whose black-gray cycle decomposition is contracted into C.

We will find it convenient to represent the cycle decomposition of G' as a graph H (Fig. 3.6c) where every cycle in C forms its own connected component and will assume that every vertex of the graph G' has two copies in H with identical labels (i.e., graph H has twice the number of vertices as compared to G'). We will show how to transform H into a breakpoint graph $G(P, Q')$ of the labelled genomes P and Q'. To achieve this goal we need to re-label the identically labelled vertices x and x in H into x_1 and x_2, and satisfy the condition that H is a breakpoint graph $G(P, Q')$ for the labelled genomes P and Q' with $\hat{Q}' = \hat{Q}$.

The genome P defines a collection of black-obverse cycles (Fig. 3.6d). Traversing black edges in graph H in the order given by these cycles defines a set of obverse edges in H (Fig. 3.6e) and a labelling of vertices in H as imposed by the fixed labelling of P. The set of these obverse edges forms matching in H and defines a gray-obverse cycle decomposition. This gray-obverse cycle decomposition defines a labelled multichromosomal genome Q' that is equivalent to Q. By the construction, the graph H with the set of obverse edges represents a breakpoint graph $G(P, Q')$ that induces the cycle decomposition C. □

3.3 Maximum Cycle Decomposition and BG-Graphs

Let $c_{max}(G')$ be the number of cycles in a maximum black-gray cycle decompositions of the contracted breakpoint graph $G' = G'(P, Q)$. Theorem 3.2.4 motivates the following reformulation of the Weak Genome Halving Problem.

Cycle Decomposition Problem. *For a given duplicated genome P, find a perfect duplicated genome Q maximizing $c_{max}(G'(P, Q))$.*

Black and gray edges of the contracted breakpoint graph $G'(P, Q)$ form a

bicolored graph that we study below.

A *BG-graph* G is a graph with black and gray edges such that the black edges form *black cycles* and the gray edges form gray matching in G (Fig. 3.6b). We refer to gray edges in G as *double* gray edges and assume that every double gray edge is a pair of parallel gray edges. This assumption implies that every BG-graph can be decomposed into edge-disjoint black-gray alternating cycles.

Below we prove an upper bound on the maximal number of black-gray cycles $c_{max}(G)$ in a cycle decomposition of the BG-graph G, and formulate necessary and sufficient conditions for achieving this bound.

A BG-graph is *connected* if it is connected with respect to the union of black and gray edges. A double gray edge in the BG-graph connecting vertices of distinct black cycles is called *interedge*. A double gray edge connecting vertices of the same black cycle is called *intra-edge*. Note that a connected BG-graph with m black cycles has at least $m-1$ interedges.

Let G be a BG-graph on $2n$ vertices with $m > 1$ black cycles, C be a black-gray cycle decomposition of G, and $e = (x, y)$ be an interedge in G. We define an *e-transformation* $(G, C) \xrightarrow{e} (G^e, C^e)$ of the graph G and its black-gray cycle decomposition C into a new BG-graph G^e on $2(n-1)$ vertices with $m-1$ black cycles and a black-gray cycle decomposition C^e of G^e of the same size as C (Fig. 3.7). In the cycle decomposition C there are two black-gray cycles c_1 and c_2 passing through edge e (it may happen that $c_1 = c_2$ when the same cycle passes through e two times). Suppose that c_1 traverses edges $(u, x), (x, y), (y, v)$ while c_2 traverses edges $(z, x), (x, y), (y, t)$. To obtain graph G^e from G we replace these triples of edges with single black edges (u, v) and (z, t) respectively and delete vertices x and y. This operation transforms the cycles c_1 and c_2 in G into into cycles c_1^e and c_2^e in G^e. We define the black-gray cycle decomposition C^e as C with the cycles c_1 and c_2 replaced with c_1^e and c_2^e.

Lemma 3.3.1. *Let C be a maximal black-gray cycle decomposition of a BG-graph G and $(G, C) \xrightarrow{e} (G^e, C^e)$ be the e-transformation for some interedge $e = (x, y)$ in G.*

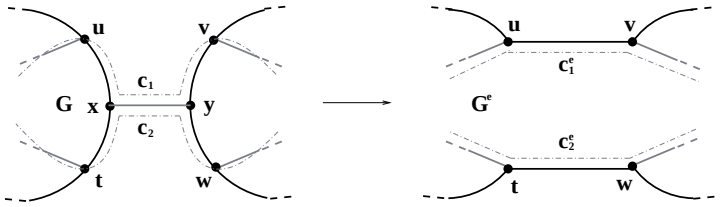

Figure 3.7 e-transformation of a graph G into a graph G^e. Black-gray cycles c_1, c_2 in G passing through interedge $e = (x, y)$ are transformed into black-gray cycles c_1^e, c_2^e in G^e. The black-gray cycle c_1 traverses edges $(u, x), (x, y), (y, v)$ that are replaced by a single edge (u, v), while black-gray cycle c_2 traverses edges $(z, x), (x, y), (y, t)$ that are replaced by a single edge (z, t). As a result, the black cycles connected by e in G are merged into a single black cycle in G^e.

Then $c_{max}(G) = c_{max}(G^e)$.

Proof. It follows from the definition of e-transformation that $c_{max}(G) = |C| = |C^e| \leq c_{max}(G^e)$. On the other hand, every black-gray cycle decomposition D^e of the graph G^e can be transformed into a black-gray cycle decomposition D of G of the same size (by simply substituting the black edges (u, v) and (z, t) in some black-gray cycles in D^e by black-gray-black triples $(u, x), (x, y), (y, v)$ and $(z, x), (x, y), (y, t)$). Therefore, $c_{max}(G^e) \leq c_{max}(G)$. □

Theorem 3.3.2. *If G is a connected BG-graph with $2n$ vertices and m black cycles, then*

$$c_{max}(G) \leq n + 2 - m = |P|/2 + 2 - m.$$

Proof. Suppose that $c_{max}(G) = k$, i.e., a maximal cycle decomposition of G contains k black-gray cycles. We find convenient to view these cycles as k disconnected cycles (i.e., every cycle forms its own connected component) that are later contracted in the BG-graph G by a series of n gluings of pairs of gray edges into double gray edges. Since one needs at least $k - 1$ such gluings to contract k disconnected black-gray cycles into a connected BG-graph, $n \geq k - 1$. It implies the theorem for $m = 1$.

Assume $m > 1$. Since the BG-graph G is connected and contains m black cycles, there exists an interedge e in G. For a maximal cycle decomposition C of the BG-graph G, consider an e-transformation $(G, C) \xrightarrow{e} (G^e, C^e)$. Lemma 3.3.1 implies

$c_{max}(G) = c_{max}(G^e)$. Note that G^e is a connected BG-graph on $2(n-1)$ vertices with $m-1$ black cycles. Iteratively applying similar e-transformations $m-1$ times we will end up with a BG-graph G^+ of size $2(n-(m-1))$ that contains a single black cycle. Hence, $c_{max}(G) = c_{max}(G^+) \leq n+2-m$. □

Note that for a BG-graph G, $c_{max}(G)$ equals the sum of $c_{max}(H)$ over all connected components H of G. Since the total size of all connected components equals $|P|$, Theorem 3.3.2 implies

$$c_{max}(G) = \sum_H c_{max}(H) \leq \sum_H |H|/2 + 2 - m_H = \\ |P|/2 + \sum_{m=1}^{\infty}(2-m)\cdot s_m \leq |P|/2 + s_1,$$

where s_m is the number of connected components with m black cycles. Let $b_e(G)$ be the number of even black cycles (i.e., black cycles of even size) in G. Note that since gray edges form a matching in BG-graph, a single odd cycle cannot form a connected component. Therefore, s_1 does not exceed $b_e(G)$,

$$c_{max}(G) \leq |P|/2 + b_e(G). \tag{3.1}$$

To achieve the upper bound (3.1), each connected component of G must contain either a single even black cycle (a *simple BG-graph*) or a pair of odd black cycles (a *paired BG-graph*). Fig. 3.6b shows a BG-graph containing an even black cycle forming a simple BG-graph, and a pair of odd black cycles forming a paired BG-graph.

We represent each black cycle of a BG-graph as points on a circle such that the arcs between adjacent points represent the black edges, and intra-edges are drawn as straight chords within these circles. A BG-graph is *non-crossing* if its intra-edges (as chords within each black circle) do not cross (Fig. 3.6b).

Theorem 3.3.3. *For a simple BG-graph G on $2n$ vertices, $c_{max}(G) = n+1$ if and only if G is non-crossing.*

Proof. We prove the theorem in both directions by induction on n. The statement is trivial for $n = 1$. Assume that the statement is true for any simple BG-graph of size $2(n-1)$ and prove it for a simple BG-graph G of size $2n$.

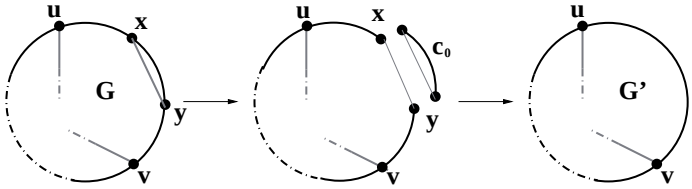

Figure 3.8 Transformation of a BG-graph G into a BG-graph G' by splitting a black-gray cycle consisting of parallel black and gray edges.

We first prove (the reasoning depends on the proof direction) that there exists a double gray edge e in G parallel to a black edge (i.e., connecting two adjacent points on a black circle) forming a black-gray cycle c_0 of length 2.

If $c_{max}(G) = n+1$, then a maximal cycle decomposition of the BG-graph G consists of $n+1$ black-gray cycles. Since these cycles contain $2n$ gray edges in total, the pigeonhole principle implies that there exists a cycle c_0 with a single gray edge e.

If the BG-graph G is non-crossing, consider a double gray edge e spanning (as a chord) the minimum number of black edges. If e spanned more than one black edge then there would exist a double gray edge with endpoints within the span of e, i.e., an edge with an even smaller span, a contradiction.

For the found edge $e = (x, y)$, let u and v be vertices adjacent to x and y on the black cycle. Transform G into a simple BG-graph G' on $2(n-1)$ vertices by removing the vertices x and y and all the incident edges, and by adding the black edge (u, v) (Fig. 3.8). Note that $c_{max}(G') = c_{max}(G) - 1$ and G' is non-crossing if and only if G is non-crossing.

By induction the graph G' is non-crossing if and only if $c_{max}(G') = n$. Therefore, G is non-crossing if and only if $c_{max}(G') = n+1$. □

Let G be a paired BG-graph G of size $2n$ (consisting of two odd black cycles) and e be an interedge in G. For a maximal black-gray cycle decomposition C of G, let $(G, C) \xrightarrow{e} (G^e, C^e)$ be an e-transformation of G. Note that the graph G^e is a simple BG-graph on $2(n-1)$ vertices. Lemma 3.3.1 and Theorem 3.3.2 imply $c_{max}(G) = c_{max}(G^e) \leq n$. Therefore, according to Theorem 3.3.3, $c_{max}(G) = n$ if and only if the BG-graph G^e is non-crossing. We are interested in a particular case of this

statement.

Theorem 3.3.4. *For a paired BG-graph G of size $2n$ with a single interedge, $c_{max}(G) = n$ if and only if G is non-crossing.*

Proof. It is easy to see that for a single interedge e in a paired BG-graph G, the e-transformation turns G into a non-crossing BG-graph if and only if G is non-crossing. □

We call a non-crossing BG-graph *primitive* if its connected components are either simple BG-graphs or paired BG-graphs with single interedges. Consequently, the contracted breakpoint graph $G'(P, Q)$ of a duplicated genome P and a perfect duplicated genome Q is called *primitive* if its black-gray subgraph is a primitive BG-graph. Theorems 3.3.3 and 3.3.4 imply

Theorem 3.3.5. *For a primitive BG-graph G, $c_{max}(G) = |P|/2 + b_e(G)$.*

A primitive BG-graph and its maximal cycle decomposition are shown at Fig. 3.6b,c.

3.4 Genome Halving Problem for Multichromosomal Genomes

3.4.1 2-Break Genome Halving Problem

In this section we solve the 2-Break Genome Halving Problem for a duplicated genome P by minimizing the 2-break distance $d_2(P, Q)$ over all perfect duplicated genomes Q. Theorems 2.2.1 and 3.2.4 motivate the following reformulation of the 2-Break Genome Halving Problem:

Cycle Decomposition Problem. *For a given duplicated (unichromosomal or multichromosomal) genome P, find a perfect duplicated (resp. unichromosomal or multichromosomal) genome Q maximizing $c_{max}(P, Q)$.*

The following theorem provides a solution to the Cycle Decomposition Problem for multichromosomal genomes:

Theorem 3.4.1. *For any duplicated genome P, there exists a perfect duplicated genome Q with $c_{max}(P,Q) = |P|/2 + b_e(P)$.*

Proof. If $\hat{P}$ contains some odd black cycles then we group them into pairs (formed arbitrarily), and introduce an arbitrary interedge connecting the cycles in each pair. We complete each black cycle with an arbitrary non-crossing gray matching. The resulting graph is a contracted breakpoint graph $G'(P,Q)$ of P and some perfect duplicated genome Q defined (not uniquely) by the double gray-obverse cycles. Since $G'(P,Q)$ is primitive, Theorem 3.3.5 implies that $c_{max}(P,Q) = |P|/2 + b_e(P)$. □

To solve the 2-Break Genome Halving Problem for a multichromosomal genome P we first find a perfect duplicated genome Q satisfying Theorem 3.4.1. Then applying Theorem 3.2.4 to a maximum black-gray cycle decomposition of $G'(P,Q)$ we get a labelling of P and some genome Q' (Q' is equivalent to Q) for which $c(P,Q') = c_{max}(P,Q') = |P|/2 + b_e(P)$. It follows from Lemma 3.2.2 that the genome Q' is a perfect duplicated genome. Theorem 3.3.5 guarantees that the decomposition of $G(P,Q')$ into $|P|/2 + b_e(P)$ black-gray cycles represents a maximal cycle decomposition, while Theorem 2.2.1 implies that it corresponds to the minimum 2-break distance between P and Q'. Therefore, the perfect duplicated genome Q' is a solution of the 2-Break Genome Halving Problem for the genome P.

3.4.2 3-Break Genome Halving Problem

In this section we solve the 3-Break Genome Halving Problem for a duplicated genome P by minimizing the 3-break distance $d_3(P,Q)$ over all perfect duplicated genomes Q. Let $c^{odd}_{max}(B)$ be the maximum number of odd black-gray cycles in a cycle decomposition among all cycle decompositions of a BG-graph B. Theorems 2.2.2 and 3.2.4 suggest the following reformulation of the 3-Break Genome Halving Problem.

Odd Cycle Decomposition Problem. *Given a duplicated genome P, find a perfect duplicated genome Q maximizing $c^{odd}_{max}(G'(P,Q))$.*

Below we use $c^{odd}_{max}(P,Q)$ as a shortcut for $c^{odd}_{max}(G'(P,Q))$. For a BG-graph B, we define $U(B)$ as

$$U(B) = \frac{|B|}{2} + b_2(B) - \frac{|b_1(B) - b_3(B)|}{2}$$

where $|B|$ is the number of black edges in B and $b_i(B)$ is the number of black cycles of length i modulo 4 in B. Later in this section (Theorem 3.4.3) we will show that for any BG-graph B, $c^{odd}_{max}(B) \leq U(B)$. Since $U(B)$ depends only on black edges in B (i.e., only on the genome P if $B = G'(P,Q)$), this inequality implies that for any perfect duplicated genome Q, $c^{odd}_{max}(P,Q) \leq \frac{|\hat{P}|}{2} + b_2(\hat{P}) - \frac{|b_1(\hat{P}) - b_3(\hat{P})|}{2}$. The following theorem shows how to achieve this upper bound.

Theorem 3.4.2. *Given a duplicated genome P, there exists a perfect duplicated genome Q with*

$$c^{odd}_{max}(P,Q) = \frac{|\hat{P}|}{2} + b_2(\hat{P}) - \frac{|b_1(\hat{P}) - b_3(\hat{P})|}{2}.$$

Proof. Genome P defines the set of black cycles in the de Bruijn graph $\hat{P}$. We will complete $\hat{P}$ with a gray matching to obtain the BG-graph $G'(P,Q)$.

First we pair every black cycle of length 1 modulo 4 with a cycle of length 3 modulo 4 (if possible) resulting in $\min\{b_1(\hat{P}), b_3(\hat{P})\}$ such pairs. The remaining $|b_1(\hat{P}) - b_3(\hat{P})|$ odd black cycles form $\frac{|b_1(\hat{P}) - b_3(\hat{P})|}{2}$ pairs arbitrarily. For each pair of odd black cycles, we introduce an arbitrary gray interedge connecting them.

For each even black cycle $(v_1, v_2, \ldots, v_{2n})$, we add n gray edges (v_1, v_2), (v_3, v_4), ..., (v_{2n-1}, v_{2n}) as shown in Fig. 3.9a. For each odd black cycle $(v_1, v_2, \ldots, v_{2n}, v_{2n+1})$ (where v_{2n+1} is incident to an interedge), we add n gray edges (v_1, v_2), (v_3, v_4), ..., (v_{2n-1}, v_{2n}) as shown in Fig. 3.9b. We remark that n gray edges $(v_1, v_2), (v_3, v_4)$, ..., (v_{2n-1}, v_{2n}) form n trivial cycles with black edges of the cycle. By Theorem 3.2.3 the resulting graph is a contracted breakpoint graph $G'(P,Q)$ for some perfect duplicated genome Q. Below we show that there exists a black-gray

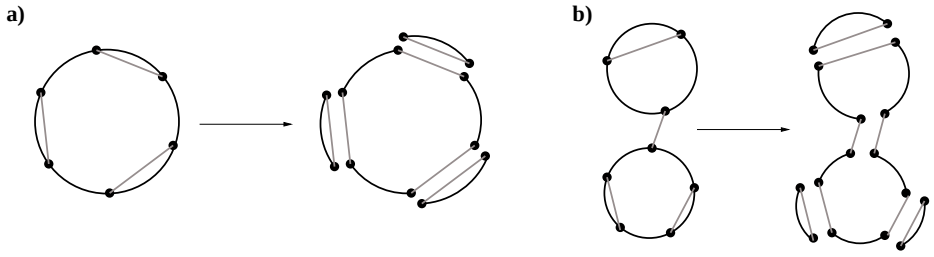

Figure 3.9 a) Cycle decomposition of a simple BG-graph on $2m$ vertices into m cycles of length 2 and one cycle of length $2m$; b) Cycle decomposition of a paired BG-graph on $2m$ vertices into $m-1$ cycles of length 2 and one cycle of length $2(m+1)$.

cycle decomposition C of the graph $G'(P,Q)$ with $c^{odd}(C) = \frac{|\hat{P}|}{2} + b_2(\hat{P}) - \frac{|b_1(\hat{P}) - b_3(\hat{P})|}{2}$ cycles.

We construct the black-gray cycle decomposition of the resulting BG-graph as follows. We decompose every even black cycle c on vertices $(v_1, v_2, \ldots, v_{2n})$ into n trivial black-gray cycles (with edges (v_1, v_2), (v_3, v_4), ..., (v_{2n-1}, v_{2n})) and one more cycle on the remaining n black edges (Fig. 3.9a). This cycle is odd iff $n = |c|/2$ is odd. Therefore, every even cycle c corresponds either to $|c|/2$ odd cycles (if $|c| = 0$ modulo 4) or to $|c|/2 + 1$ odd cycles (if $|c| = 2$ modulo 4).

Similarly, each paired component p formed by odd cycles $(v_1, v_2, \ldots, v_{2n+1})$ and $(w_1, w_2, \ldots, w_{2m+1})$ can be decomposed into $n + m$ trivial black-gray cycles (formed by edges (v_1, v_2), (v_3, v_4), ..., (v_{2n-1}, v_{2n}) and (w_1, w_2), (w_3, w_4), ..., (w_{2m-1}, w_{2m})) and one more "large" cycle on the remaining $n + m + 2$ black edges of the component (Fig. 3.9b). This "large" cycle is odd iff $n + m + 2 = |p|/2 + 1$ is odd. Therefore, every paired component p corresponds either to $|p|/2$ odd cycles (if $|p| = 0$ modulo 4) or to $|p|/2 - 1$ odd cycles (if $|p| = 2$ modulo 4).

Therefore, each component with n black edges is decomposed into $n/2$ odd cycles unless it is an even cycle of length 2 modulo 4 (in this case it is one odd cycle more) or a paired component of size 2 modulo 4 (in this case it is one odd cycle less). Summing over all connected components we get $\frac{|\hat{P}|}{2} + b_2(\hat{P}) - \frac{|b_1(\hat{P}) - b_3(\hat{P})|}{2}$ cycles. □

The rest of this section is devoted to the proof of the following theorem and the outline of the 3-Break Halving Algorithm.

Theorem 3.4.3. *For any BG-graph B, $c^{odd}_{max}(B) \leq U(B)$.*

Proof. We first give a sketch of the proof to provide an intuition for the follow up Lemmas 3.4.4-3.4.8.

We prove Theorem 3.4.3 by induction on the number of interedges $\iota(B)$ in B. Lemma 3.4.4 proves the base case of $\iota(B) = 0$. For any BG-graph B with $\iota(B) > 0$ and its black-gray cycle decomposition C with the maximum number of odd black-gray cycles (i.e., $c^{odd}(C) = c^{odd}_{max}(B)$) we show how to transform it into a BG-graph B' with a black-gray cycle decomposition C' such that $\iota(B') < \iota(B)$, $U(B') \leq U(B)$, and $c^{odd}(C') \geq c^{odd}(C)$. Then by induction

$$c^{odd}_{max}(B) = c^{odd}(C) \leq c^{odd}(C') \leq U(B') \leq U(B).$$

The construction of such a pair (B', C') breaks into two cases depending on whether every interedge in B is shared by two distinct odd cycles from C or not. The latter case is addressed by Lemmas 3.4.5 and 3.4.6, while the former case is addressed by Lemmas 3.4.7 and 3.4.8. □

Lemma 3.4.4. *For a simple BG-graph B, $c^{odd}_{max}(B) \leq U(B)$.*

Proof. For a simple BG-graph B, $b_1(B) = b_3(B) = 0$ and $U(B) = \frac{|B|}{2} + b_2(B)$. The inequality (3.1) implies $c^{odd}_{max}(B) \leq c_{max}(B) \leq |B|/2 + b_e(B) = |B|/2 + 1$. If $|B| = 2$ modulo 4 then $U(B) = |B|/2 + 1$ and $c^{odd}_{max}(B) \leq U(B)$. If $|B| = 0$ modulo 4 then $U(B) = |B|/2$ while $c^{odd}_{max}(B) \leq |B|/2 + 1$. However, in this case the inequality $c^{odd}_{max}(B) \leq |B|/2 + 1$ is not tight since the overall number of odd cycles in every cycle decomposition of a simple BG-graph is even while $|B|/2 + 1$ is odd. Therefore, $c^{odd}_{max}(B) \leq |B|/2 = U(B)$. □

Our proof of Theorem 3.4.3 is based on the notion of e-transformations introduced in Section 3.3. For a double gray edge $e = (x, y)$ in B, e-transformation transforms the BG-graph B and its black-gray cycle decomposition C into a new BG-graph B^e with a black-gray cycle decomposition C^e as follows. Let c_1 and c_2 be two cycles (that may coincide) from C sharing the double gray edge $e = (x, y)$ and suppose

that the cycle c_1 (resp. c_2) passes through the vertices u, x, y, v (resp. z, x, y, t) in a row. Recall that the BG-graph B^e is defined as the BG-graph B with the vertices x, y and all the incident edges replaced with two new black edges (u, v) and (z, t) (Fig. 3.7). A black-gray cycle decomposition C^e of the graph B^e is obtained from C by replacing u, x, y, v in the cycle c_1 with a single black edge (u, v) and replacing z, x, y, t in the cycle c_2 with a single black edge (z, t).

Note that if e is an interedge then e-transformation eliminates the interedge e and "merges" two black cycles in B into a single cycle in B^e (Fig. 3.7). Since such merging cannot create new interedges, we have $\iota(B^e) \leq \iota(B) - 1$. In the following two Lemmas we study how e-transformations affect the parameters $U(B)$ and $c^{odd}(C)$.

Lemma 3.4.5. *If e is an interedge in B then e-transformation does not increase $U(B)$, i.e, $U(B^e) \leq U(B)$.*

Proof. Every e-transformation reduces $|B|$ by two and may change each of the expressions $b_2(B)$ and $\frac{|b_1(B)-b_3(B)|}{2}$ by at most 1. However, if e-transformation increases $b_2(B)$ by 1 then it cannot change $\frac{|b_1(B)-b_3(B)|}{2}$ thus implying that $U(B^e) = \frac{|B^e|}{2} + b_2(B^e) - \frac{|b_1(B^e)-b_3(B^e)|}{2} \leq \frac{|B|-2}{2} + (b_2(B)+1) - \frac{|b_1(B)-b_3(B)|}{2} = U(B)$. Indeed, if $b_2(B)$ increases (i.e., $b_2(B^e) = b_2(B)+1$) then e-transformation creates a new cycle of length 2 modulo 4 implying that the interedge e connects black cycles whose lengths sum up to 0 modulo 4. If their lengths are 1 and 3 modulo 4 then $\frac{|b_1(B)-b_3(B)|}{2}$ does not change, and if they both are of even length then $\frac{|b_1(B)-b_3(B)|}{2}$ does not change either. □

Lemma 3.4.6. *Let C be a cycle decomposition of a BG-graph B and e be an interedge shared by cycles c_1 and c_2 from C. Then e-transformation does not reduce $c^{odd}(C)$ (i.e., $c^{odd}(C^e) \geq c^{odd}(C)$) unless c_1 and c_2 are two distinct odd cycles (in this case $c^{odd}(C^e) = c^{odd}(C) - 2$).*

Proof. If $c_1 = c_2$ (i.e., cycles c_1 and c_2 are the same) then e-transformation simply reduces the number of black edges in this cycle by 2, i.e., $c^{odd}(C^e) = c^{odd}(C)$. If c_1 and c_2 are distinct cycles then e-transformation reduces the number of black edges

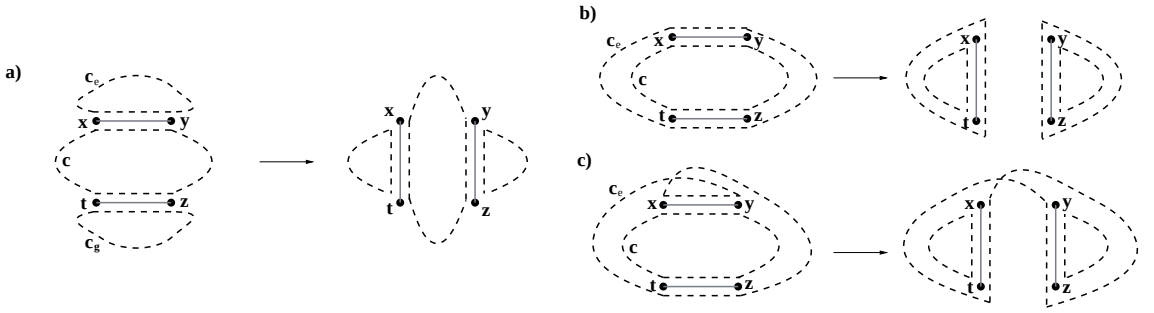

Figure 3.10 Types of (e, g)-transformations operating on double gray edges $e = (x, t)$ and $g = (z, t)$ of cycle c: a) cycles c_e and c_g are different; b) cycle $c_e = c_g$ traverses vertices x, y, z, t as $(\ldots, x, y, \ldots, z, t)$; c) cycle $c_e = c_g$ traverses vertices x, y, z, t as $(\ldots, x, y, \ldots, t, z)$.

in each of these cycles by 1. Therefore, $c^{odd}(C)$ may reduce by 2, increase by 2 or remain the same. The reduction happens only if both c_1 and c_2 are odd cycles. □

As soon as there is an interedge e in a BG-graph B that does not belong to two distinct odd cycles in a black-gray cycle decomposition C, Lemmas 3.4.5 and 3.4.6 allow one to perform the induction step in the proof of Theorem 3.4.3. To analyze cycle decompositions with every interedge shared by two distinct odd cycles, we introduce (e, g)-*transformations* of BG-graphs that replace a pair of gray edges $e = (x, y)$ and $g = (z, t)$ belonging to the same cycle c from C with a pair of gray edges (y, z) and (x, t). There may be up to two more cycles in C containing the gray edges e and g: c_e (cycle containing e) and c_g (cycle containing g). The (e, g)-transformation splits cycle c and transforms cycles c_e and c_g as follows.

If $c_e \neq c_g$, cycles c_e and c_g are merged into a single cycle (Fig. 3.10a). If $c_e = c_g$ then there are two possibilities (Fig. 3.10b,c) depending on how c_e traverses edges e and g: either as $(\ldots, x, y, \ldots, z, t)$ or as $(\ldots, x, y, \ldots, t, z)$. In the former case c_e is split into two cycles (Fig. 3.10b) while in the latter case it is rearranged (Fig. 3.10c).

In summary, (e, g)-transformation $(B, C) \rightarrow (B^{(e,g)}, C^{(e,g)})$, may either merge cycles c_e and c_g (if $c_e \neq c_g$), or rearrange/split them (if $c_e = c_g$). Note that since B and $B^{(e,g)}$ have the same black subgraph, $U(B^{(e,g)}) = U(B)$.

Lemma 3.4.7. *Let C be a black-gray cycle decomposition of a BG-graph B where a double gray edge e is shared by two distinct even black-gray cycles. Then there exists an (e,g)-transformation with $U(B^{(e,g)}) = U(B)$, $c^{odd}(C^{(e,g)}) \geq c^{odd}(C) + 2$, and $\iota(B^{(e,g)}) \leq \iota(B)$.*

Proof. Let c and c_e be two even black-gray cycles in C containing double gray edge $e = (x, y)$. Let $g = (z, t)$ be the "next" double gray edge in c after e (i.e., (y, z) form a black edge in B) and c_g be a cycle in C sharing the edge g with c. Consider an (e, g)-transformation $(B, C) \longrightarrow (B^{(e,g)}, C^{(e,g)})$.

Note that among two new double gray edges (x, t) and (y, z) only the double gray edge (x, t) may be an interedge in $B^{(e,g)}$. Moreover, (x, t) is an interedge in $B^{(e,g)}$ iff either (x, y) or (z, t) is an interedge in B. Therefore, $\iota(B^{(e,g)}) \leq \iota(B)$.

Note that (e, g)-transformation splits c into two odd cycles increasing the number of odd cycles by 2. Note also that (e, g)-transformation either merges c_e and c_g into a cycle $c_e + c_g$ (in case $c_e \neq c_g$) or splits/rearranges c_e (in case $c_e = c_g$). Since c_e is even then in the former case $c_e + c_g$ is odd iff c_g is odd while in the latter case the number of odd cycle can only increase. Therefore, $c^{odd}(C^{(e,g)}) \geq c^{odd}(C) + 2$. □

Lemma 3.4.8. *Let B be a BG-graph with $\iota(B) > 0$ and C be its black-gray cycle decomposition such that every interedge is shared by two distinct odd cycles from C. Then there exist a BG-graph B' with black-gray cycle decomposition C' such that $U(B') \leq U(B)$, $c^{odd}(C') \geq c^{odd}(C)$, and $\iota(B') < \iota(B)$.*

Proof. Let $e = (x, y)$ be an interedge in B and c be an odd black-gray cycles from C passing through e. Cycle c has at least two interedges and let $g = (z, t)$ be the "next" interedge in cycle c after e (i.e., there is no other interedges between y and z while travelling along c, implying that y and z belong to the same black cycle in B). Note that since e is shared by two distinct odd cycles, $g \neq e$.

Consider an (e, g)-transformation $(B, C) \longrightarrow (B^{(e,g)}, C^{(e,g)})$ that replaces (x, y) and (z, t) with (y, z) and (x, t). This transformation removes two interedges from B and introduce at most one interedge (since (y, z) is not an interedge in $B^{(e,g)}$),

implying $\iota(B^{(e,g)}) < \iota(B)$.

The (e, g)-transformation splits the odd cycle c into an even cycle (which we denote c') and an odd cycle. Hence, such splitting does not affect the number of odd cycles. We now analyze how the (e, g)-transformation affects odd cycles c_e and c_g (that are different from c) passing through edges e and g, correspondingly.

If $c_e = c_g$ then the (e, g)-transformation either rearranges this odd cycle (preserving the length) or splits it into two. In either case, that does not affect the number of odd cycles, i.e., $c^{odd}(C^{(e,g)}) = c^{odd}(C)$. Therefore, letting $B' = B^{(e,g)}$ and $C' = C^{(e,g)}$ proves the theorem.

If $c_e \neq c_g$ then these odd cycles are merged into an even cycle c'' in $C^{(e,g)}$ implying that $c^{odd}(C^{(e,g)}) = c^{odd}(C)-2$. But in this case the even cycles c' and c'' share a double gray edge (i.e., either (x, t) or (y, z)). By Lemma 3.4.7 there exist a BG-graph B' and its black-gray cycle decomposition C' such that $c^{odd}(C') \geq c^{odd}(C^{(e,g)}) + 2 = c^{odd}(C)$, $U(B') = U(B)$, and $\iota(B') < \iota(B)$. □

This completes the proof of Theorem 3.4.3. We now outline the linear-time 3-Break Genome Halving Algorithm:

1. For a given duplicated genome P, find a perfect duplicated genome Q such that $c^{odd}_{max}(P, Q) = |P|/2 + b_2(\hat{P}) - \frac{|b_1(\hat{P})-b_3(\hat{P})|}{2}$ and a maximum black-gray cycle decomposition C of the graph $G'(P, Q)$ (Theorem 3.4.2).

2. Find a labelling of the genomes P and Q' (Q' is equivalent to Q) and a breakpoint graph $G(P, Q')$ inducing C (Theorem 3.2.4). Output Q' as a solution of the 3-Break Genome Halving Problem.

3.5 Genome Halving Problem for Unichromosomal Genomes

We first outline the differences between the Genome Halving Problems for unichromosomal and multichromosomal genomes. The following theorem gives a characterization of the contracted breakpoint graphs for unichromosomal genomes (compare to Theorem 3.2.1).

Theorem 3.5.1. *A graph H with black, gray, and obverse edges is a contracted breakpoint graph for some duplicated genomes if and only if*

- *each vertex in H is incident to two black edges, two gray edges, and a pair of parallel obverse edges;*
- *H is connected with respect to the union of black and obverse edges (*black-obverse connected*);*
- *H is connected with respect to the union of gray and obverse edges (*gray-obverse connected*).*

Proof. Suppose that graph H is a contracted breakpoint graph of the genomes P and Q (represented as black-obverse P-cycle and gray-obverse Q-cycle). The graph H is simply the result of gluing these P-cycle and Q-cycle. Since gluing cycles cannot disconnect them, the graph H is both black-obverse and gray-obverse connected.

Consider a black-obverse and gray-obverse connected graph H where each vertex is incident to two black edges, two gray edges, and a pair of parallel obverse edges. Label endpoints of each obverse edge x in H by x^t and x^h. Since the graph H is black-obverse connected, there exists an alternating Eulerian black-obverse cycle traversing all black and obverse edges in this graph. The order of vertices in this cycle defines some duplicated genome P. Similarly, since the graph H is gray-obverse connected, there exists an alternating Eulerian gray-obverse cycle traversing all gray and obverse edges that defines some duplicated genome Q. Then the graph H is a contracted breakpoint graph for the genomes P and Q. □

Labelling Problem

Lemma 3.5.2. *The perfect duplicated (unichromosomal) genome $R \oplus R$ is equivalent to the two-chromosomal genome $2R$. Moreover, $2R$ is the only genome equivalent to $R \oplus R$.*

Proof. It is easy to see that the gluing of both $R \oplus R$ and $2R$ represented as gray-obverse cycles results in a single gray-obverse cycle c that traverses R in order (every

edge in this cycle has multiplicity 2) (Fig. 3.3b,c). Any other genome that is glued into c cannot have a cycle shorter than c since such a short cycle would remain short after gluing. This implies that every genome that is glued into c either traverses c twice ($R \oplus R$) or is formed by two cycles each of which traverses c once ($2R$). □

Theorem 3.2.4 and Lemma 3.5.2 imply the following theorem (compare to Theorem 3.2.4):

Theorem 3.5.3. *Let P and $R \oplus R$ be unichromosomal duplicated genomes and C be a black-gray cycle decomposition of the contracted breakpoint graph $G'(P, R \oplus R)$. Then there exists some labelling of either $R \oplus R$ or $2R$ that induces the cycle decomposition C.*

Theorem 3.5.3 reveals the connection between the Weak Genome Halving Problem and maximal cycle decomposition and breaks the analysis of the Weak Genome Halving Problem into two cases depending on whether the maximal cycle decomposition is induced by $R \oplus R$ or $2R$.

Cycle Decomposition Problem

In order to solve the Cycle Decomposition Problem for a given genome P, we will construct a contracted breakpoint graph $G'(P, R \oplus R)$ which achieves the upper bound (3.1). The de Bruijn graph $\hat{P}$, being a subgraph of $G'(P, R \oplus R)$ (for any pre-duplicated genome R), completely defines a vertex set, an obverse matching, and a set of black cycles in G' (Fig. 3.3a,d,e). We will show how to complete the graph $\hat{P}$ with a set of double gray edges to obtain a contracted breakpoint $G'(P, R \oplus R)$ with the maximum value of $c_{max}(G')$.

A *BO-graph* is a connected graph with black and obverse edges such that the black edges form black cycles and the obverse edges form an obverse matching (every duplicated genome P corresponds to a BO-graph $\hat{P}$). A *BOG-graph* is a graph with black, obverse, and gray edges where black and obverse edges form a BO-graph (a *BO-subgraph*), and black and gray edges form a primitive BG-graph

(a *BG-subgraph*). Note that each black-gray connected component of a BOG-graph is a simple non-crossing BG-graph or a paired non-crossing BG-graph with a single interedge.

The arguments above suggest that the Cycle Decomposition Problem for a genome P can be reformulated as follows. For a given BO-graph G (defined as $G = \hat{P}$), find a gray-obverse connected BOG-graph G' having G as a BO-subgraph. Theorems 3.5.1 and 3.3.5 imply that such a BOG-graph graph is a contracted breakpoint graph $G'(P, R \oplus R)$ for some genome R for which $c_{max}(G')$ achieves the upper bound (3.1).

We remark that gray-obverse connected components of a BOG-graph form gray-obverse cycles (alternating double gray and obverse edges). Hence, a BOG-graph is gray-obverse connected if and only if it has a single gray-obverse cycle.

Lemma 3.5.4. *For a BOG-graph with more than one gray-obverse cycle, there exists a black edge connecting two different gray-obverse cycles.*

Proof. Let H be a BOG-graph with two or more gray-obverse cycles. Since H is black-obverse connected there exists a black-obverse cycle in H traversing all obverse edges of H. Therefore, there exists a black edge connecting obverse edges from different gray-obverse cycles in H. □

Theorem 3.5.5. *For a given BO-graph G, there exists a BOG-graph G' with a single gray-obverse cycle having G as a BO-subgraph.*

Proof. First we group odd black cycles in G into pairs (formed arbitrarily), and introduce an arbitrary interedge connecting cycles in each pair. Then we complete each black cycle with an arbitrary non-crossing gray matching so that each vertex of G becomes incident to exactly one double gray edge. Denote the resulting graph by H. Note that H is a BOG-graph having G as a BO-subgraph.

If H has a single gray-obverse cycle, then the theorem holds for $G' = H$. Otherwise, we show how to modify the set of double gray edges in H to reduce the number of gray-obverse cycles.

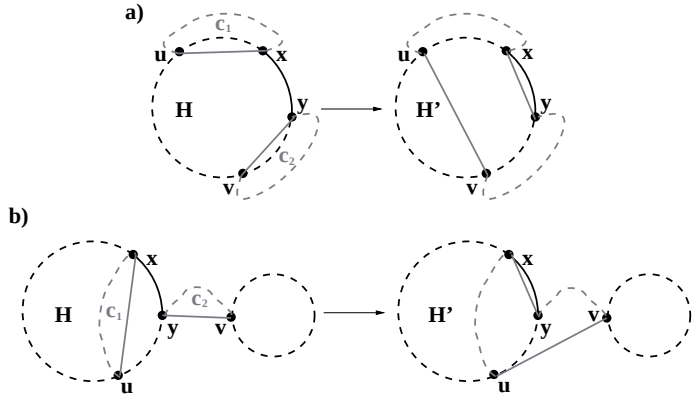

Figure 3.11 Merging gray-obverse cycles c_1, c_2 connected by a black edge (x, y) passing through a) intra-edges (x, u) and (y, v); b) an intra-edge (x, u) and an interedge (y, v).

Assume that there is more than one gray-obverse cycle in H. By Lemma 3.5.4 there is a black edge (x, y) connecting distinct gray-obverse cycles c_1 and c_2. Let (x, u) and (y, v) be double gray edges incident to the vertices x and y respectively. We replace the edges (x, u) and (y, v) in H with double gray edges (x, y) and (u, v) resulting in a graph H'. Fig. 3.11 illustrates two cases depending on whether the edge (y, v) is an interedge (since (x, u) and (y, v) belong to the same black-gray connected component, at most one of them can be an interedge).

We will show that the BG-subgraph of H' is primitive. There are two new double gray edges in the BG-subgraph of H' compared to H. Since the introduced double gray edge (x, y) is parallel to a black edge, it does not cross any other intra-edge (as chords). The introduced double gray edge (u, v) is either an intra-edge or an interedge. In the former case any intra-edge crossing the intra-edge (u, v) would necessary cross (x, u) or (y, v) (as chords), a contradiction to the fact that H has a non-crossing BG-subgraph. Hence, the BG-subgraph of H' is non-crossing. On the other hand, it is easy to see that the transformation $H \rightarrow H'$ turns a simple black-gray connected component of the graph H into a simple black-gray connected component of H' (Fig. 3.11a), and a paired black-gray connected component with a single interedge into a paired black-gray connected component with a single interedge (Fig. 3.11b). Hence, the BG-subgraph of H' is primitive and H' is a BOG-graph.

Note that the BOG-graph H' has G as a BO-subgraph (since black and

obverse edges were not affected by the transformation). The graph H' has the same gray-obverse cycles as H, except for the gray-obverse cycles c_1 and c_2 which are joined into a single cycle in H'. Hence, the number of gray-obverse cycles in H' is reduced as compared to H.

Iteratively reducing the number of gray-obverse cycles we will eventually come up with a BOG-graph G' having G as a BO-subgraph with a single gray-obverse cycle. □

Theorems 3.3.5 and 3.5.5 imply the following theorem (compare to Theorem 3.4.1):

Theorem 3.5.6. *For any duplicated genome P, there exists a perfect duplicated genome $R \oplus R$ such that $c_{max}(P, R \oplus R) = |P|/2 + b_e(P)$, and each paired component of $G'(P, R \oplus R)$ contains a single interedge.*

Although the maximal black-gray cycle decomposition of $G'(P, R \oplus R)$ may correspond to a breakpoint graph $G(P, 2R)$ (Fig. 3.5), we will prove below that there exists a breakpoint graph $G(P, R \oplus R)$ having "almost" the same number of black-gray cycle as $G(P, 2R)$ (Fig. 3.5d). In Section 3.5.2 we will classify all the cases where there exists a labelled genome $R' \oplus R'$ such that $c(P, R' \oplus R') = c(P, 2R)$.

Graphs $G'(P, R \oplus R)$ and $G'(P, 2R)$

Lemma 3.5.2 implies that $G'(P, R \oplus R) = G'(P, 2R)$ for any duplicated genome P. But in difference from the breakpoint graph $G(P, R \oplus R)$ (for any labelling of P and $R \oplus R$) that contains a single gray-obverse cycle, the breakpoint graph $G(P, 2R)$ contains two gray-obverse cycles. The following theorem reveals the relationship between $G(P, R \oplus R)$ and $G(P, 2R)$.

Theorem 3.5.7. *For any labellings of the genomes P and $2R$, there exists a labelling of the genome $R \oplus R$ such that $|c(P, R \oplus R) - c(P, 2R)| \leq 1$. Moreover, if there are two gray edges (x, y) and $(\bar{x}, \bar{y})$ belonging to the same black-gray cycle in $G(P, 2R)$ then there exists a labelling of $R \oplus R$ with $c(P, R \oplus R) \geq c(P, 2R)$.*

Proof. Let (x, y) be a gray edge in the breakpoint graph $G(P, 2R)$. Since the genome $2R$ is perfect duplicated there exists a gray edge $(\bar{x}, \bar{y})$ connecting counterparts of x and y. Define a graph H having the same vertices and edges as $G(P, 2R)$ except the gray edges (x, y) and $(\bar{x}, \bar{y})$ that are replaced with the gray edges $(x, \bar{y})$ and $(\bar{x}, y)$. Since the graph $G(P, 2R)$ consists of two gray-obverse cycles, the gray edges (x, y) and $(\bar{x}, \bar{y})$ belong to different gray-obverse cycles. Therefore, the graph H contains a single gray-obverse cycle (as well as a single black-obverse cycle inherited from $G(P, 2R)$). This implies that H is a breakpoint of the labelled genomes P and $R \oplus R$ (where the labelling of P is the same as in $G(P, 2R)$).

If the gray edges (x, y) and $(\bar{x}, \bar{y})$ belong to the same black-gray cycle in $G(P, 2R)$ then this cycle may be split into two in H while the other black-gray cycles are not affected. Conversely, if the gray edges (x, y) and $(\bar{x}, \bar{y})$ belong to different black-gray cycles in $G(P, 2R)$ then these cycles may be joined into a single cycle in H. In either case the difference $|c(P, R \oplus R) - c(P, 2R)|$ does not exceed 1. □

3.5.1 A Flaw in El-Mabrouk–Sankoff Analysis

El-Mabrouk and Sankoff came up with a theorem describing the minimum distance from the given rearranged duplicated genome to a perfect duplicated genome. Given a rearranged duplicated genome P, the crux of their approach is an algorithm for computing $c(G)$ – the number of cycles of so-called *maximal completed graph*, i.e., a breakpoint graph[3] with the maximum number of black-gray cycles. In [24] they demonstrate that $c(G)$ equals the number of genes plus $\gamma(G)$ where $\gamma(G)$ is the parameter defined below. We illustrate the concepts from [24] using the genome $P = +a + b - c + b - d - e + a + c - d - e$ on the set of genes $\mathcal{B} = \{a, b, c, d, e\}$ (p. 757 in [24]). El-Mabrouk and Sankoff first arbitrarily label two copies of each gene x as x_1 and x_2 for each $x \in \mathcal{B}$ and further transform the signed permutation G into an unsigned permutation $a_1^t a_1^h b_1^t b_1^h c_1^h c_1^t b_2^t b_2^h d_1^h d_1^t e_1^h e_1^t a_2^t a_2^h c_2^t c_2^h d_2^h d_2^t e_2^h e_2^t$.

Let $\mathbf{V} = \{x_1^t, x_1^h, x_2^t, x_2^h \mid x \in \mathcal{B}\}$. The *partial graph* $\mathcal{G}(\mathbf{V}, A)$ associated with

[3]Following El-Mabrouk and Sankoff [24], we ignore obverse edges in breakpoint graphs throughout Section 3.5.1.

P has the edge set A of black edges linking adjacent term (other than obverses x_i^t and x_i^h) in the corresponding unsigned permutation (Fig. 3.12a).

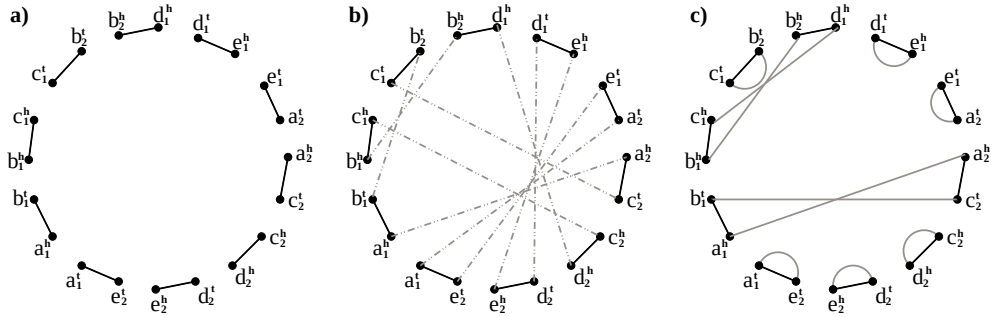

Figure 3.12 a) A set of black edges forming the partial graph $\mathcal{G}(\mathbf{V}, A)$ corresponding to the genome $P = +a + b - c + b - d - e + a + c - d - e$; b) Natural graphs as connected components in the partial graph with counterpart edges; c) A completed graph $\mathcal{G}(\mathbf{V}, A, \Gamma)$ with maximum number of cycles $c(G) = 8$. $\mathcal{G}(\mathbf{V}, A, \Gamma)$ is a breakpoint graph of the circular genome $P = +a_1 + b_1 - c_1 + b_2 - d_1 - e_1 + a_2 + c_2 - d_2 - e_2$ and a perfect duplicated genome $(-a_1 + e_2 + d_2 - c_2 + b_1)(-b_2 + c_1 - d_1 - e_1 + a_2)$ (of the form $R \ominus R$).

Black edges together with counterpart edges (i.e., edges between x_1^t and x_2^t or between x_1^h and x_2^h) form a graph shown in Fig. 3.12b. The connected components of this graph are called *natural graphs* in [24]. There are four connected components (natural graphs) in the graph in Fig. 3.12b, two of them have 3 black edges (odd natural graphs) and two of them have 2 black edges (even natural graphs). Let NE be the number of even natural graphs ($NE = 2$ in Fig. 3.12b).

El-Mabrouk and Sankoff define the parameter

$$\gamma(G) = \begin{cases} NE, & \text{if all natural graphs are even} \\ NE + 1, & \text{otherwise} \end{cases}$$

A graph $\mathcal{G}(\mathbf{V}, A, \Gamma)$ obtained from the partial graph $\mathcal{G}(\mathbf{V}, A)$ by introducing a set of gray edges Γ is called a *completed graph* if $\mathcal{G}(\mathbf{V}, A, \Gamma)$ is a breakpoint graph for some genomes on the set of genes $\{x_1, x_2 \mid x \in \mathcal{B}\}$. The following theorem (Theorem 7.7 in [24]) characterizes the maximum number of cycles in the completed graph $\mathcal{G}(\mathbf{V}, A, \Gamma)$.

Theorem. *The maximal number of cycles in a completed graph of* $\mathcal{G}(\mathbf{V}, A)$ *is* $c(G) =$

$\frac{|A|}{2} + \gamma(G)$.

For the genome in Fig. 3.12 we have $\gamma(G) = NE + 1 = 3$ and $c(G) = \frac{|A|}{2} + \gamma(G) = \frac{10}{2} + 3 = 8$. A completed graph $\mathcal{G}(\mathbf{V}, A, \Gamma)$ with 8 cycles is shown at Fig. 3.12c.[4] Below we provide a counterexample to Theorem 7.7 from [24].

Consider a circular genome $P = +a+b-a-b$ labelled as $+a_1+b_1-a_2-b_2$. The genome P defines a partial graph $\mathcal{G}(\mathbf{V}, A)$ with a single natural graph of even size, implying $\gamma(G) = 1$. It follows from Theorem 7.7 in [24] that there exists a perfect duplicated genome Q such that the breakpoint graph $G = G(P, Q)$ consists of $\frac{|A|}{2} + \gamma(G) = 3$ cycles. However, the direct enumeration of all possible perfect duplicated genomes Q shows that there is no breakpoint graph $G(P, Q)$ with 3 cycles. There exist eight distinct labelled perfect duplicated genomes Q giving rise to eight breakpoint graphs $G(P, Q)$ shown in Fig. 3.13. All of them have less than 3 cycles. In the next section we explain what particular property of the genome $+a+b-a-b$ was not addressed properly in the El-Mabrouk–Sankoff analysis.

3.5.2 Classification Of Unichromosomal Duplicated Genomes

To introduce a new combinatorial invariant of duplicated genomes, consider labellings of vertices in the cycle defined by the duplicated rearranged genome P with numbers 0 and 1 (Fig. 3.14b). Every such labelling induces a two-digit labelling of the genes (edges): a label of each gene is formed by the labels of the incident vertices (Fig. 3.14c). A 01-labelling of the vertices is called *consistent* if for every pair of identical genes in P the label of one copy is inversion of the other. If there exist consistent labellings of genome P, we define the *parity index* of P as the number of genes labelled "01" modulo 2. Below we prove that the parity index is well-defined, i.e., the parity index is the same for all consistent labellings of a genome. It turns out that the El-Mabrouk–Sankoff theorem fails on genomes with the parity index 0.

We re-define the notion of parity of a genome P in terms of the de Bruijn graph $\hat{P}$. A genome P is called *singular* if all black cycles in $\hat{P}$ are even. For

[4] While we do not explicitly consider $R \ominus R$ duplications shown in this Figure (see [24] for details), our counterexample works for both $R \oplus R$ and $R \ominus R$ duplications.

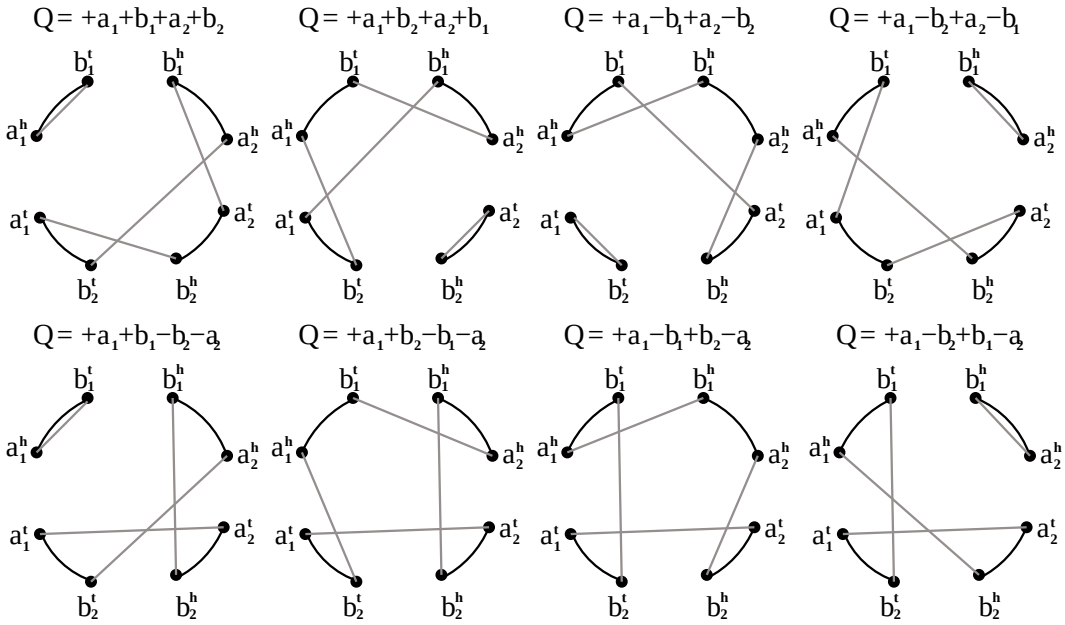

Figure 3.13 Breakpoint graphs of the circular genome $P = +a+b-a-b$ and all possible labellings of all possible perfect duplicated genomes Q (without loss of generality we assume that the labelling of P as $+a_1 + b_1 - a_2 - b_2$ is fixed). In terms of [24], the top four graphs correspond to $R \oplus R$ duplication pattern while the bottom four graphs correspond to to $R \ominus R$ duplication pattern.

a non-singular genome P, define $\mathit{parity}(P) = \infty$. For a singular genome P, we clockwise label edges of each black cycle in $\hat{P}$ with alternating numbers $\{0, 1\}$ so that every two adjacent edges are labelled differently (Fig. 3.15a). Labels of black edges in cycle P classify obverse edges in P into two classes: *even* if its flanking black edges have the same labels, and *odd* if its flanking black edges have different labels (Fig. 3.15b). Let m_{even} and m_{odd} be the number of even/odd obverse edges in P correspondingly. Obviously, both m_{even} and m_{odd} are even numbers. We define $\mathit{parity}(P) = m_{odd}/2 \bmod 2$.

This definition of the parity index coincides with the one given in the beginning of this section. To establish a correspondence between them one can consider a genome P as a black-obverse cycle and contract each black edge into a single vertex that inherits the label from the black edge. Since every pair of adjacent black edges of $\hat{P}$ is labelled differently, every pair of counterpart vertices is labelled differently as well. This implies that two-digit labels of every pair of obverse edges are inversions

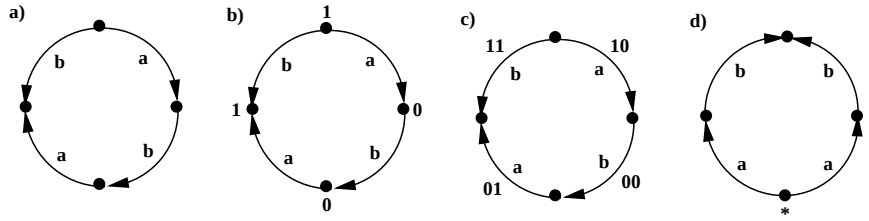

Figure 3.14 a) Circular genome $P = +a-b+a+b$ represented as a cycle with directed edges; b) 01-labelling of the vertices of the cycle defined by P; c) Induced labelling of the genes of P that is consistent; d) For some genomes consistent labellings do not exist: for genome $Q = +a+b-b-a$ the labels of both copies of gene a start with the same digit ("star") so they cannot be inversion of each other.

of each other.

Theorem 3.5.8. *The parity index of a singular genome is well defined.*

Proof. Let P be a singular genome. If $\hat{P}$ has k black cycles then there are 2^k different 01-labellings of its black edges (two possible labellings per cycle). Therefore, it is sufficient to show that a change of 01-labelling of a particular black cycle c does not affect $parity(P)$.

Let m^c_{even} and m^c_{odd} be the number of even/odd obverse edges in cycle P connecting black edges of c with black edges outside c. Since double obverse edges form matching in the de Bruijn graph $\hat{P}$, the total number of double obverse edges connecting c with other black cycles is even and, thus, $m^c_{even} + m^c_{odd}$ is a multiple of 4.

Change of 01-labelling of the black cycle c reverses the labels $0 \leftrightarrow 1$ in c. Reversed labelling of c does not change parity of obverse edges connecting two black edges in c (since both endpoint labels change) or two black edges outside of c (since neither of endpoint labels change). At the same time, each of $m^c_{even} + m^c_{odd}$ obverse edges connecting black edges in c with black edges outside c changes its parity (i.e., even edges become odd and vice versa). Then m_{odd} changes into m'_{odd} equal to:

$$m_{odd} - m^c_{odd} + m^c_{even} = m_{odd} - (m^c_{odd} + m^c_{even}) + 2m^c_{even}$$

Since both $m^c_{odd} + m^c_{even}$ and $2m^c_{even}$ are multiples of 4, the parity of $m'_{odd}/2$ and $m_{odd}/2$ is the same implying that $parity(P)$ is well defined. □

Our goal is to prove the following theorem:

Theorem 3.5.9. *For a duplicated genome P,*

$$\max_R c(P, R \oplus R) = \begin{cases} |P|/2 + b_e(P), & \textit{if } \textit{parity}(P) \neq 0 \\ |P|/2 + b_e(P) - 1, & \textit{otherwise} \end{cases}$$

The proof of Theorem 3.5.9 is split into two cases depending on whether P is singular or non-singular.

Theorem 3.5.10. *For a non-singular genome P, $\max_R c(P, R \oplus R) = |P|/2 + b_e(P)$.*

Proof. If P is a non-singular genome then $\hat{P}$ has an odd black cycle. According to Theorem 3.5.6 there exists a perfect duplicated genome $R \oplus R$ such that $G'(P, R \oplus R)$ is primitive and $c_{max}(P, R \oplus R) = |P|/2 + b_e(P)$. Theorem 3.5.3 ensures that the maximum cycle decomposition of the contracted breakpoint graph $G'(P, R \oplus R)$ is induced by a labelling of either $R \oplus R$ or $2R$. If it is $R \oplus R$ then the theorem holds. Otherwise, consider a paired component in $G'(P, R \oplus R)$ (which exists since $\hat{P}$ has an odd black cycle) and an interedge e in it. Let (x, y) and $(\bar{x}, \bar{y})$ be gray edges in $G(P, 2R)$ corresponding to the interedge e in $G'(P, 2R) = G'(P, R \oplus R)$. Since $G'(P, R \oplus R)$ is primitive and e is the only bridge between two different black cycles in $G'(P, R \oplus R)$, the gray edges (x, y) and $(\bar{x}, \bar{y})$ must belong to the same black-gray cycle in $G(P, 2R)$. Applying Theorem 3.5.7 to these gray edges we obtain a labelled genome $R \oplus R$ with $c(P, R \oplus R) = c(P, 2R) = |P|/2 + b_e(P)$. □

For a singular genome P, we first fix some alternating 01-labelling of black edges in every black cycle of $\hat{P}$. The labelling of edges imposes a labelling of vertices of any breakpoint graph $G(P, Q)$ (for any genome Q) so that each vertex inherits a label from an incident black edge. Note that every pair of counterpart vertices get different labels as their incident black edges are adjacent in $\hat{P}$. A labelling of vertices of $G(P, Q)$ is called *uniform* if endpoints of every gray edge have identical labels (i.e, every gray edge is even). We will need the following theorem:

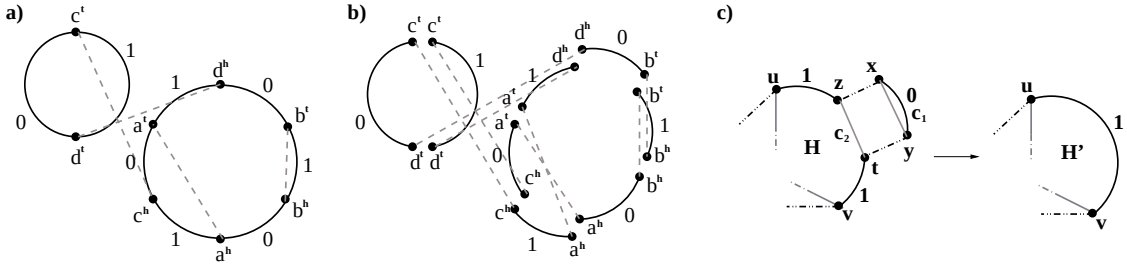

Figure 3.15 For the genome $P = +a-b-b-d+c-a-d+c$, a) 01-labelling of the de Bruijn graph $\hat{P}$; b) induced labelling of the black-obverse cycle P with $m_{odd} = 4$ and $m_{even} = 4$; c) transformation of the graph H into H' by removing vertices x, y, z, t and incident edges and adding a black edge (u, v) labeled the same as (u, x) and (v, t).

Theorem 3.5.11. *Let P be a singular genome and Q be a perfect duplicated genome with $c(P, Q) = |P|/2 + b_e(P)$. Then every alternating 01-labelling of $\hat{P}$ imposes a uniform labelling on vertices of $G(P, Q)$.*

While the definition of the breakpoint graph does not explicitly specify the counterpart edges, one can derive them for $G(P, Q)$ in Theorem 3.5.11 from the vertex labels. Also, it is easy to see that gray and counterpart edges in $G(P, Q)$ form cycles of length 4 as soon as Q is a perfect duplicated genome. We take a liberty to restate the condition $c(P, Q) = |P|/2 + b_e(P)$ as $c_{bg}(G) = n + c_{bc}(G)$, where $c_{bg}(G)$ is the number of black-gray edges in G, n is the number of unique genes in P and $c_{bc}(G)$ is the number of black-counterpart cycles in G. Also, every alternating 01-labelling of $\hat{P}$ corresponds to an alternating labelling of black edges within black-counterpart cycles. This leads to the following reformulation of Theorem 3.5.11:

Theorem 3.5.12. *Let H be a graph on $4n$ vertices consisting of three perfect matchings: black, gray, and counterpart such that (i) gray and counterpart matchings form cycles of length 4 and (ii) $c_{bg}(H) = n + c_{bc}(H)$. Then every alternating 01-labelling of black edges within black-counterpart cycles imposes a uniform labelling on vertices of H.*

Proof. The proof is done by induction on n. If $n = 1$ then the graph H consists of a gray-counterpart cycle with two black edges parallel to the gray edges, and the theorem holds. Assume that the theorem holds for graphs with less than $4n$ vertices.

Since H has $2n$ black edges and $c_{bg}(H) = n + c_{bc}(H) > n$, the pigeonhole principle implies that there exists a trivial black-gray cycle c_1 in H. Let $e_1 = (x, y)$ be a gray edge in the cycle c_1 (thus, e_1 is even) and let (x, z) and (y, t) be adjacent counterpart edges. Then there is a gray edge $e_2 = (z, t)$ belonging to the same gray-counterpart cycle as e_1. Let c_2 be a black-gray cycle c_2 containing the gray edge e_2.

If the cycle c_2 is trivial, then the endpoints of e_2 have identical labels. In this case we define a new graph H' as the graph H without vertices x, y, z, t and all incident edges. It is easy to see that H' is a graph on $4(n-1)$ vertices satisfying the conditions of the theorem. Indeed, the number of black-gray cycles in H' is reduced by 2 and the number of black-counterpart cycles is reduced by 1 (as compared to H), i.e., $c_{bg}(H') = c_{bg}(H) - 2$ and $c_{bc}(H') = c_{bc}(H) - 1$. Therefore, $c_{bg}(H') = (n-1) + c_{bc}(H')$. By the induction assumption, every alternating 01-labelling of H' imposes a uniform labelling on vertices of H'. It implies that every alternating 01-labelling of H imposes a uniform labelling on vertices of H.

If the cycle c_2 is not trivial, let (u, z) and (t, v) be black edges adjacent to e_2. These black edges are neighbors of the black edge (x, y) on a black-counterpart cycle (passing through the vertices u, z, x, y, t, v), so they have the same label l which different from the label of (x, y). Therefore, the endpoints of the gray edge e_2 have identical labels. We define a new graph H' as the graph H with vertices x, y, z, t and all incident edges removed but with a single black edge (u, v) labelled l added (Fig. 3.15c). The graph H' has $4(n-1)$ vertices, $c_{bc}(H') = c_{bc}(H)$ black-counterpart cycles, and $c_{bg}(H') = c_{bg}(H) - 1$ black-gray cycles, thus, $c_{bg}(H') = n - 1 + c_{bc}(H')$ and the induction applies. □

To complete the proof of Theorem 3.5.9 we need one more theorem:

Theorem 3.5.13. *For a singular genome P and a perfect duplicated genome Q with $c(P, Q) = |P|/2 + b_e(P)$,*

- *$Q = R \oplus R$ iff parity$(P) = 1$;*

- $Q = 2R$ *iff* $parity(P) = 0$.

Proof. According to Theorem 3.5.3, the graph $G(P, Q)$ has either a single gray-obverse cycle (case $Q = R \oplus R$) or two symmetric gray-obverse cycles (case $Q = 2R$). Theorem 3.5.11 implies that all gray edges in $G(P, Q)$ are even (i.e., have identically labelled endpoints) for every alternating 01-labelling of black edges of P.

Case 1: Graph $G(P, Q)$ has a single gray-obverse cycle c. Consider an arbitrary vertex v in $G(P, Q)$ and its counterpart $\overline{v}$. Vertices v and $\overline{v}$ break c into two paths: c' (from v to $\overline{v}$) and c'' (from $\overline{v}$ to v). For every path (cycle) c denote c_{odd} as the number of odd obverse edges in c. Note that obverse edges are evenly divided between c' and c'', i.e., for every pair of obverse edges connecting counterpart vertices, one edge belongs to c' and the other edge belongs to c''. Therefore, $c'_{odd} = c''_{odd}$. Note that start (vertex v) and end (vertex $\overline{v}$) vertices of path c' are labelled differently. Since the total number of odd edges is odd for every path with differently labelled ends and since all gray edges are even (Theorem 3.5.11), the total number of odd obverse edges in the path c' is odd. Therefore, $c_{odd}/2 = c'_{odd}$ is odd implying that $parity(P) = 1$.

Case 2: Graph $G(P, Q)$ has two gray-obverse cycles c' and c''. Note that obverse edges are evenly divided between c' and c'', i.e., for every pair of obverse edges connecting counterpart vertices, one edge belongs to c' and the other edge belongs to c''. Therefore, $c'_{odd} = c''_{odd}$. Since the total number of odd edges in every cycle is even and since all gray edges are even (Theorem 3.5.11), the total number of odd obverse edges in every cycle is even. Since c'_{odd} is even, the overall number of odd obverse edges is a multiple of 4 implying that $parity(P) = 0$. □

For a singular genome P with $parity(P) = 1$ Theorem 3.5.13 implies Theorem 3.5.9 while for a singular genome P with $parity(P) = 0$ it implies that there is no genome R for which $c(P, R \oplus R) = |P|/2 + b_e(P)$. In the latter case, there exists a genome R and a labelling of P and $2R$ for which $c(P, 2R) = |P|/2 + b_e(P)$ (Theorems 3.5.6 and 3.5.3). The genome $2R$ can be transformed into a labelled genome $R \oplus R$ with $c(P, R \oplus R) = c(P, 2R) - 1 = |P|/2 + b_e(P) - 1$ (Theorem 3.5.7). This

completes the proof of Theorem 3.5.9.

3.5.3 Genome Halving Algorithm

The results of previous sections lead to the following algorithm for Genome Halving Problem[5]:

1. For a given duplicated genome P, find a perfect duplicated genome $R \oplus R$ such that $G'(P, R \oplus R)$ is primitive and decompose $G'(P, R \oplus R)$ into $c_{max}(P, R \oplus R) = |P|/2 + b_e(P)$ black-gray cycles (Theorem 3.5.6).

2. Find a labelling of the genomes P and Q ($Q = R \oplus R$ or $Q = 2R$) and a breakpoint graph $G(P, Q)$ inducing the maximum black-gray cycle decomposition of $G'(P, R \oplus R)$ (Theorem 3.5.3).

3. If $Q = R \oplus R$ then output the breakpoint graph $G(P, R \oplus R)$.

4. If $Q = 2R$ and P is non-singular then there is a paired component in $G'(P, R \oplus R)$ with a single interedge (since $G'(P, R \oplus R)$ is primitive) that corresponds to two gray edges (x, y) and $(\bar{x}, \bar{y})$ in $G(P, 2R)$. Find a labelling of the genome $R \oplus R$ for which $c(P, R \oplus R) = c(P, 2R)$ (Theorems 3.5.7 and 3.5.10) and output $G(P, R \oplus R)$.

5. If $Q = 2R$ and P is singular then $parity(P) = 0$ (Theorem 3.5.13). Find a labelling of the genome $R \oplus R$ for which $c(P, R \oplus R) = c(P, 2R) - 1$ (Theorem 3.5.7) and output $G(P, R \oplus R)$.

To estimate the complexity of the Genome Halving Algorithm we assume that every graph is implemented as a collection of sets: a set of vertices, sets of edges of each color, and an array of sets of incident edges indexed by vertices and colors. Note that for a given genome P with n genes, all graphs appearing in the algorithm have vertex and edge sets of order $O(n)$ while every set of incident edges contains

[5]The algorithm below outputs the breakpoint graph $G(P, R \oplus R)$ (in addition to the pre-duplicated genome R). This allows one to reconstruct a sequence of reversals transforming $R \oplus R$ into P with the reversal distance algorithm.

at most 2 elements. Therefore, even with a straightforward data structure each set operation (such as an insertion/deletion of an element or a membership query) takes $O(n)$ time.

One can demonstrate that every step of the Genome Halving Algorithm can be done in $O(n)$ set operations. Therefore, the overall time complexity of the Genome Halving Algorithm can be estimated as $O(n^2)$. In practice, our implementation of the Genome Halving Algorithm takes less than a second to halve a "random" duplicated genome with 1000 unique genes with a standard Intel PIII-900MHz CPU.

Acknowledgements

This chapter is based on the following three papers:

- Max A. Alekseyev and Pavel A. Pevzner. "Whole Genome Duplications and Contracted Breakpoint Graphs". *SIAM Journal on Computing*, 2007, 36(6), pp. 1748-1763.
- Max A. Alekseyev and Pavel A. Pevzner. "Colored de Bruijn Graphs and the Genome Halving Problem". *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2007, 4, pp. 98-107.
- Max A. Alekseyev and Pavel A. Pevzner. "Whole Genome Duplications, Multi-Break Rearrangements, and Genome Halving Theorem". *Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2007, pp. 665-679.

The dissertation author was the primary investigator and author of these papers.

4 Conclusions

4.1 Summary of Contributions

Multi-Break Rearrangements

Multi-break rearrangements represent a generalization of the standard rearrangement operations (i.e., reversals, translocations, fusions, fissions) as well as transpositions. We demonstrated that the multi-breaks are easier to analyze than the standard rearrangement operations, came up with an explicit formula for computing the multi-break distance between two circular genomes, and proposed a linear-time algorithm (that requires generation of certain Hilbert and Gröbner bases) for computing this distance. We contributed to the ongoing debates on "RBM vs. FBM" controversy by analyzing multi-break rearrangements in mammalian evolution and demonstrating that even if complex rearrangements like transpositions were a dominant force, the Pevzner–Tesler arguments against RBM still stand. We also demonstrated the theoretical advantages of multi-break rearrangements (as compared to the standard rearrangements and transpositions) in solving problems that may be intractable otherwise (e.g., the 3-Break Genome Halving Problem). We further extended some of these results from the case of circular genomes to the much harder case of linear genomes.

Duplicated Genomes and the Genome Halving Problem

We proposed an alternative approach to the Genome Halving Problem based on the new notion of the *contracted breakpoint graph* which is a generalization of the

conventional breakpoint graph to the case of duplicated genomes. As the breakpoint graph plays an important role in computing the rearrangement distance between non-duplicated genomes (the Hannenhalli–Pevzner theory), the contracted breakpoint graph provides important insights into computing the rearrangement distance between duplicated genomes. Using the contracted breakpoint graph, we made a number of contributions to the Genome Halving Problem. In particular, we gave a new proof of the original El-Mabrouk–Sankoff Genome Halving theorem for multichromosomal circular genomes. We also identified a flaw in the original El-Mabrouk–Sankoff Genome Halving theorem for unichromosomal circular genomes and fixed this flaw by introducing a new combinatorial invariant defined on duplicated circular permutations. This led to an effective classification of all genomes for which the El-Mabrouk–Sankoff theorem does not hold and to the new Genome Halving theorem that adequately deals with all genomes. We further proceeded to solving a novel 3-Break Genome Halving Problem for rearrangements involving more complex transposition-like operations.

4.2 Future Research

A number of interesting questions are left open for future research.

Halving of yeast genomes

While it is easy to solve the Genome Halving Problem for any given duplicated genome, it is not clear which of many possible solutions is biologically adequate. Currently we are exploring an approach that attempts to "guide" the process of genome halving (*Guided Genome Halving Problem*), using certain auxiliary information in order to restrict the number of possible solutions and to relate them more closely to the specifics of particular genomes. In guided halving of the *S. cerevisiae* genome (represented as a mosaic of blocks from the *K. waltii* genome with each block appearing two times [37]) the auxiliary information comes from yet another yeast genome *A. gossypii* with an 1-to-2 mapping into *S. cerevisiae* [19].

Searching for still unknown duplication events

The original approach to proving that one genome is a duplicated version of another genome (by establishing a 1-to-2 correspondence as in [37]) inspired a quest for similar arguments for other (pairs of) genomes. Even when applied to relatively distant genomes, it may reveal interesting and unexplored connections. Using similar tiling of the genomes we plan to come up with an universal measure of duplicativity (*duplicativity index*) and to develop a tool for computing it. For genomes with high duplicativity index (similarly to the *S. cerevisiae* yeast genome), we plan to investigate whether they undergone the whole genome duplication or not. It would be also challenging to determine and to analyze duplicated regions across multiple genomes, and to see whether there are duplication hotspots (i.e., genes that are more likely being duplicated than the others).

Rearrangement distance between duplicated genomes

There is a number of open problems related to computing the rearrangement distance between duplicated genomes. One of them is a long standing problem of computing the rearrangement distance between genomes with each gene appearing in exactly two copies. We believe that the contracted breakpoint graph is a powerful tool for the rearrangement analysis of duplicated genomes, that has a potential to address this problem. Using the contracted breakpoint graph, computing the rearrangement distance between duplicated genomes can be posed as a graph-theoretic problem. This reduction may help to resolve the complexity status of this problem as well as to come up with a good approximation algorithm.

Bibliography

[1] D. A. Bader, B. M. E. Moret, and M. Yan. A linear-time algorithm for computing inversion distances between signed permutations with an experimental study. *J. Comput. Biol.*, 8:483–491, 2001.

[2] M. Bader and E. Ohlebusch. Sorting by weighted reversals, transpositions, and inverted transpositions. *Proceedings of the 10th Conference on Research in Computational Molecular Biology (RECOMB)*, pages 563–577, 2006.

[3] V. Bafna and P. A. Pevzner. Sorting permutations by transpositions. *SIAM J. Discrete Math.*, 11:224–240, 1998.

[4] J. Bailey, R. Baertsch, W. Kent, D. Haussler, and E. Eichler. Hotspots of mammalian chromosomal evolution. *Genome Biology*, 5(4):R23, 2004.

[5] E. Belda, A. Moya, and F. J. Silva. Genome rearrangement distances and gene order phylogeny in γ-proteobacteria. *Mol. Biol. Evol.*, 22:1456–1467, 2005.

[6] A. Bergeron. A very elementary presentation of the Hannenhalli–Pevzner theory. *Lecture Notes in Computer Science*, 2089:106–117, 2001.

[7] A. Bergeron, J. Mixtacki, and J. Stoye. Reversal distance without hurdles and fortresses. *Lecture Notes in Computer Science*, 3109:388–399, 2004.

[8] Anne Bergeron, Julia Mixtacki, and Jens Stoye. A Unifying View of Genome Rearrangements. *Lecture Notes in Computer Science*, 4175:163–173, 2006.

[9] Cecile Neuveglise Jacky de Montigny Michel Aigle Francois Artiguenave Gaelle Blandin Monique Bolotin-Fukuhara Elisabeth Bon Philippe Brottier Serge Casaregola Pascal Durrens Claude Gaillardin Andree Lepingle Odile Ozier-Kalogeropoulos Serge Potier William Saurin Fredj Tekaia Claire Toffano-Nioche Micheline Wesolowski-Louvel Patrick Wincker Jean Weissenbach Jean-Luc Souciet Bertrand Llorente, Alain Malpertuy and Bernard Dujon. Genomic exploration of the hemiascomycetous yeasts: 18. comparative analysis of chromosome maps and synteny with saccharomyces cerevisiae. *FEBS Letters*, 487(1):101–112.

[10] G. Bourque, P. A. Pevzner, and G. Tesler. Reconstructing the genomic architecture of ancestral mammals: lessons from human, mouse, and rat genomes. *Genome Research*, 14:507–516, 2004.

[11] G. Bourque, Y. Yacef, and N. El-Mabrouk. Maximizing synteny blocks to identify ancestral homologs. *Lecture Notes in Bioinformatics*, 3678:21–34, 2005.

[12] G. Bourque1, E. M. Zdobnov, P. Bork, P. A. Pevzner, and G. Tesler. Comparative architectures of mammalian and chicken genomes reveal highly variable rates of genomic rearrangements across different lineages. *Genome Research*, 15:98–110, 2005.

[13] Alberto Caprara. On the Tightness of the Alternating-Cycle Lower Bound for Sorting by Reversals. *Journal of Combinatorial Optimization*, 3(2-3):149–182, 1999.

[14] X. Chen, J. Zheng, P. Nan Z. Fu, Y. Zhong, S. Lonardi, and T. Jiang. Computing the assignment of orthologous genes via genome rearrangement. *Proceedings of Asia Pacific Bioinformatics Conference*, pages 363–378, 2005.

[15] D. A. Christie. *Genome Rearrangement Problems.* PhD thesis, University of Glasgow, 1999.

[16] A. Christoffels, E. G. L. Koh, J. Chia, S. Brenner, S. Aparicio, and B. Venkatesh. Fugu genome analysis provides evidence for a whole-genome duplication early during the evolution of ray-finned fishes. *Mol. Biol. Evol.*, 21(6):1146–1151, 2004.

[17] D. Cox, J. Little, and D. O'Shea. *Ideals, varieties, and algorithms.* Springer-Verlag, 1996.

[18] P. Dehal and J. L. Boore. Two rounds of genome duplication in the ancestral vertebrate genome. *PLoS Biology*, 3(10):e314, 2005.

[19] F. S. Dietrich et al. The *Ashbya gossypii* Genome as a Tool for Mapping the Ancient *Saccharomyces cerevisiae* Genome. *Science*, 304:304–307, 2004.

[20] N. El-Mabrouk. Genome Rearrangement by Reversals and Insertions/Deletions of Contiguous Segments. *Lecture Notes in Computer Science*, 1848:222–234, 2000.

[21] N. El-Mabrouk, B. Bryant, and D. Sankoff. Reconstructing the pre-doubling genome. *Proceedings of the Third Annual International Conference on Computational Molecular Biology (RECOMB)*, pages 154–163, 1999.

[22] N. El-Mabrouk, J. H. Nadeau, and D. Sankoff. Genome Halving. *Lecture Notes in Computer Science*, 1448:235–250, 1998.

[23] N. El-Mabrouk and D. Sankoff. On the reconstruction of ancient doubled circular genomes. *Genome Informatics*, 10:83–93, 1999.

[24] N. El-Mabrouk and D. Sankoff. The Reconstruction of Doubled Genomes. *SIAM Journal on Computing*, 32:754–792, 2003.

[25] I. Elias and T. Hartman. A 1.375-Approximation Algorithm for Sorting by Transpositions. *Lecture Notes in Computer Science*, 3692:204–214, 2005.

[26] G.-M. Greuel, G. Pfister, and H. Schönemann. SINGULAR 3.0.2. A Computer Algebra System for Polynomial Computations, Centre for Computer Algebra, University of Kaiserslautern, 2006. `http://www.singular.uni-kl.de`.

[27] Q. P. Gu, S. Peng, and H. Sudborough. A 2-approximation algorithm for genome rearrangements by reversals and transpositions. *Theoret. Comput. Sci.*, 210:327–339, 1999.

[28] R. Guyot and B. Keller. Ancestral genome duplication in rice. *Genome*, 47:610–614, 2004.

[29] S. Hannenhalli and P. Pevzner. Transforming men into mouse (polynomial algorithm for genomic distance problem). *Proceedings of the 36th Annual Symposium on Foundations of Computer Science*, pages 581–592, 1995.

[30] S. Hannenhalli and P. Pevzner. Transforming cabbage into turnip (polynomial algorithm for sorting signed permutations by reversals). *Journal of the ACM*, 46:1–27, 1999.

[31] T. Hartman. A simpler 1.5-approximation algorithm for sorting by transpositions. *Lecture Notes in Computer Science*, 2676:156–169, 2003.

[32] T. Hartman and R. Sharan. A 1.5-approximation algorithm for sorting by transpositions and transreversals. *Lecture Notes in Computer Science*, 3240:50–61, 2004.

[33] H. Hinsch and S. Hannenhalli. Recurring genomic breaks in independent lineages support genomic fragility. *BMC Evolutionary Biology*, 6:90, 2006.

[34] O. Jaillon et al. Genome duplication in the teleost fish *Tetraodon nigroviridis* reveals the early vertebrate proto-karyotype. *Nature*, 431:946–957, 2004.

[35] H. Kaplan, R. Shamir, and R.Tarjan. Faster and simpler algorithm for sorting signed permutations by reversals. *SIAM Journal on Computing*, 29:880–892, 1999.

[36] H. Kaplan and E. Verbin. Sorting signed permutations by reversals, revisited. *J. Comput. Syst. Sci.*, 70(3):321–341, 2005.

[37] M. Kellis, B. W. Birren, and E. S. Lander. Proof and evolutionary analysis of ancient genome duplication in the yeast *Saccharomyces cerevisiae*. *Nature*, 428:617–624, 2004.

[38] Hiroshi Kikuta, Mary Laplante, Pavla Navratilova, Anna Z Komisarczuk, Par G. Engstrom, David Fredman, Altuna Akalin, Mario Caccamo, Ian Sealy, Kerstin Howe, Julien Ghislain, Guillaume Pezeron, Philippe Mourrain, Staale Ellingsen, Andrew C. Oates, Christine Thisse, Bernard Thisse, Isabelle Foucher, Birgit Adolf, Andrea Geling, Boris Lenhard, and Thomas S. Becker. Genomic regulatory blocks encompass multiple neighboring genes and maintain conserved synteny in vertebrates. *Genome Research*, 17(5):545–555, 2007.

[39] R. Koszul, S. Caburet, B. Dujon, and G. Fischer. Eucaryotic genome evolution through the spontaneous duplication of large chromosomal segments. *The EMBO Journal*, 23:234–243, 2004.

[40] R. B. Langkjaer, M. L. Nielsen, P. R. Daugaard, W. Liu, and J. Piskur. Yeast chromosomes have been significantly reshaped during their evolutionary history. *Journal of Molecular Biology*, 304(3):271–288.

[41] D. Levy, M. Vazquez, M. Cornforth, B. Loucas, R. K. Sachs, and J. Arsuaga. Comparing DNA damage-processing pathways by computer analysis of chromosome painting data. *J. Comput. Biol.*, 11:626–641, 2004.

[42] G. H. Lin and G. Xue. Signed genome rearrangements by reversals and transpositions: models and approximations. *Theoret. Comput. Sci.*, 259:513–531, 2001.

[43] Y. C. Lin, C. L. Lu, H.-Y. Chang, and C. Y. Tang. An Efficient Algorithm for Sorting by Block-Interchanges and Its Application to the Evolution of Vibrio Species. *J. Comput. Biol.*, 12:102–112, 2005.

[44] Michael R. Mehan, Maricel Almonte, Erin Slaten, Nelson B. Freimer, P. Nagesh Rao, and Roel A. Ophoff. Analysis of segmental duplications reveals a distinct pattern of continuation-of-synteny between human and mouse genomes. *Human Genetics*, 121(1):93–100, 2007.

[45] A. Meyer and Y. Van de Peer. From 2R to 3R: evidence for a fish-specific genome duplication (FSGD). *BioEssays*, 27(9):937–945, 2005.

[46] W. J. Murphy, G. Bourque, G. Tesler, P. Pevzner, and S. J. O'Brien. Reconstructing the genomic architecture of mammalian ancestors using multispecies comparative maps. *Human Genomics*, 1:30–40, 2003.

[47] W. J. Murphy, D. M. Larkin, A. Everts van der Wind, G. Bourque, G. Tesler, L. Auvil, J. E. Beever, B. P. Chowdhary, F. Galibert, L. Gatzke, C. Hitte, C. N. Meyers, D. Milan, E. A. Ostrander, G. Pape, H. G. Parker, T. Raudsepp, M. B. Rogatcheva, L. B. Schook, L. C. Skow, M. Welge, J. E. Womack, S. J. O'Brien, P. A. Pevzner, and H. A. Lewin. Dynamics of Mammalian Chromosome

Evolution Inferred from Multispecies Comparative Map. *Science*, 309(5734):613–617, 2005.

[48] J. H. Nadeau and B. A. Taylor. Lengths of Chromosomal Segments Conserved since Divergence of Man and Mouse. *Proceedings of the National Academy of Sciences*, 81(3):814–818, 1984.

[49] S. Ohno. *Evolution by gene duplication*. Springer, Berlin, 1970.

[50] M. Ozery-Flato and R. Shamir. Two Notes on Genome Rearrangement. *Journal of Bioinformatics and Computational Biology*, 1:71–94, 2003.

[51] D. V. Pasechnik. On computing the Hilbert bases via the Elliott–MacMahon algorithm. *Theoretical computer science*, 263:37–46, 2001. Implementation: `http://stuwww.uvt.nl/~dpasech/software.html`.

[52] Q. Peng, P. A. Pevzner, and G. Tesler. The Fragile Breakage versus Random Breakage Models of Chromosome Evolution. *PLoS Computational Biology*, 2:e14, 2006.

[53] P. Pevzner, H. Tang, and G. Tesler. De Novo Repeat Classification and Fragment Assembly. *Genome Research*, 14:1786–1796, 2004.

[54] P. Pevzner and G. Tesler. Genome Rearrangements in Mammalian Evolution: Lessons from Human and Mouse Genomes. *Genome Research*, 13(1):37–45, 2003.

[55] P. A. Pevzner. *Computational Molecular Biology: An Algorithmic Approach*. The MIT Press, Cambridge, 2000.

[56] P. A. Pevzner and G. Tesler. Human and mouse genomic sequences reveal extensive breakpoint reuse in mammalian evolution. *Proceedings of the National Academy of Sciences*, 100:7672–7677, 2003.

[57] A. J. Radcliffe, A. D. Scott, and E. L. Wilmer. Reversals and Transpositions Over Finite Alphabets. *SIAM J. Discrete Math.*, 19:224–244, 2005.

[58] A. Ruiz-Herrera, J. Castresana, and T. J. Robinson. Is mammalian chromosomal evolution driven by regions of genome fragility? *Genome Biology*, 7:R115, 2006.

[59] R. K. Sachs, J. Arsuaga, M. Vazquez, L. Hlatky, and P. Hahnfeldt. Using graph theory to describe and model chromosome aberrations. *Radiat Research*, 158:556–567, 2002.

[60] R. K. Sachs, D. Levy, P. Hahnfeldt, and L. Hlatky. Quantitative analysis of radiation-induced chromosome aberrations. *Cytogenetic and Genome Research*, 104:142–148, 2004.

[61] D. Sankoff. Genome rearrangement with gene families. *Bioinformatics*, 15:909–917, 1999.

[62] D. Sankoff. The signal in the genome. *PLoS Computational Biology*, 2(4):0320–0321, 2006.

[63] D. Sankoff and P. Trinh. Chromosomal breakpoint re-use in the inference of genome sequence rearrangement. *Proceedings of the Eighth Annual International Conference on Computational Molecular Biology (RECOMB)*, pages 30–35, 2004.

[64] D. Sankoff and P. Trinh. Chromosomal Breakpoint Reuse in Genome Sequence Rearrangement. *Journal of Computational Biology*, 12(6):812–821, 2005.

[65] Lucy Skrabanek and Kenneth H. Wolfe. Eukaryote genome duplication - where's the evidence? *Current Opinion in Genetics & Development*, 8(6):694–700, 1998.

[66] K. M. Swenson, M. Marron, J. V. Earnest-DeYoung, and B. M. E. Moret. Approximating the true evolutionary distance between two genomes. *Proc. 7th Workshop on Algorithm Engineering & Experiments (ALENEX)*, pages 121–129, 2005.

[67] K. M. Swenson, N. D. Pattengale, and B. M. E. Moret. A framework for orthology assignment from gene rearrangement data. *Lecture Notes in Bioinformatics*, 3678:153–166, 2005.

[68] E. Tannier and M.-F. Sagot. Sorting by reversals in subquadratic time. *Lecture Notes in Computer Science*, pages 1–13, 2004.

[69] G. Tesler. Efficient algorithms for multichromosomal genome rearrangements. *J. Comput. Syst. Sci.*, 65:587–609, 2002.

[70] A. Everts van der Wind, S. R. Kata, M. R. Band, M. Rebeiz, D. M. Larkin, R. E. Everts, C. A. Green, L. Liu, S. Natarajan, T. Goldammer, J. H. Lee, S. McKay, J. E. Womack, and H. A. Lewin. A 1463 Gene Cattle-Human Comparative Map With Anchor Points Defined by Human Genome Sequence Coordinates. *Genome Research*, 14(7):1424–1437, 2004.

[71] M. Vazquez et al. Computer analysis of mFISH chromosome aberration data uncovers an excess of very complicated metaphases. *Int. J. Radiat. Biol.*, 78(12):1103–1115, 2002.

[72] M. E. Walter, Z. Dias, and J. Meidanis. Reversal and transposition distance of linear chromosomes. *String Processing and Information Retrieval: A South American Symposium (SPIRE)*, pages 96–102, 1998.

[73] M. E. Walter, L. Reginaldo, A. F. Curado, and A. G. Oliveira. Working on the Problem of Sorting by Transpositions on Genome Rearrangements. *Lecture Notes in Computer Science*, 2676:372–383, 2003.

[74] C. Webber and C. P. Ponting. Hotspots of mutation and breakage in dog and human chromosomes. *Genome Research*, 15(12):1787–1797, 2005.

[75] Kenneth H. Wolfe and Denis C. Shields. Molecular evidence for an ancient duplication of the entire yeast genome. *Nature*, 387:708–713, 1997.

[76] S. Yancopoulos, O. Attie, and R. Friedberg. Efficient sorting of genomic permutations by translocation, inversion and block interchange. *Bioinformatics*, 21:3340–3346, 2005.

[77] Y. Yue and T. Haaf. 7E olfactory receptor gene clusters and evolutionary chromosome rearrangements. *Cytogenetic and Genome Research*, 112:6–10, 2006.

[78] S. Zhao, J. Shetty, L. Hou, A. Delcher, B. Zhu, K. Osoegawa, P. de Jong, W. C. Nierman, R. L. Strausberg, and C. M. Fraser. Human, Mouse, and Rat Genome Large-Scale Rearrangements: Stability Versus Speciation. *Genome Research*, 14:1851–1860, 2004.